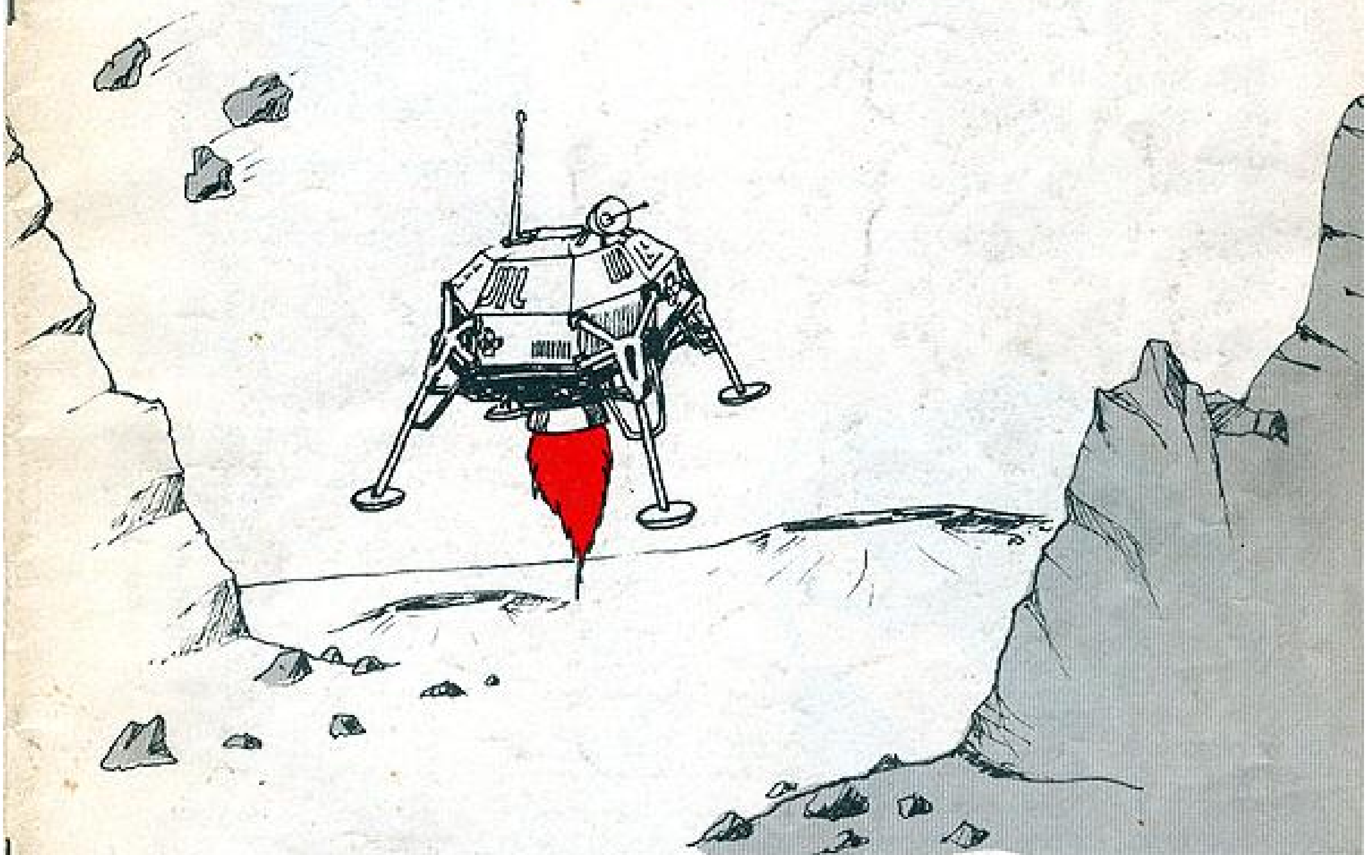
Vol. 1, Issue 2

Price $2.00

ROM

The Magazine That Brings The ATARI™ Computer to Life!



- TITAN LANDER (MACHINE LANGUAGE)
- STAR BOMB
- SCOTT ADAMS INTERVIEW
- RAVING REVIEWER
- YELLOW BRICK ROAD
- SCROLLING YOUR ATARI

SPECIAL SPACE GAME ISSUE

ATARI SINGS YOUR FAVORITE SONGS!!!

THE Original VOICE BOX Speech Synthesizer by the ALIEN GROUP has received rave reviews:

MICRO COMPUTING–"The VOICE BOX injects an endearing personality to your computer. The possibilities are enormous."

COMPUTE–"The VOICE BOX offers more human-like tones and does not blank out the screen."

CREATIVE COMPUTING–"English text and phonetic code may be freely intermixed rather than requiring separate modes as is the case without exception with every other speech system. A mode called talking face displays an animated face with impressive lip sync animation."

ANTIC–"There is a great potential for teaching children to spell and an added dimension to games overall. I believe the VOICE BOX is well worth the price tag."

ANALOG–"For ATARI owners who want to add speech to their programs, the Alien Group VOICE BOX is probably the best choice."

POPULAR SCIENCE–"The speech quality is excellent. Besides creating speech, the software has a bit of fun with graphics."

and on the new VOICE BOX II......

TIME MAGAZINE–"Machine of the Year" "The VOICE BOX by the Alien Group enables an ATARI to say aloud anything typed on its keyboard in any language. It also sings "Amazing Grace" and "When I'm 64" or anything else that anyone wants to teach it."

VOICE BOX II
Speech & Singing Synthesizer

To order by mail send a check or money order to the ALIEN GROUP for $169. Then, try the VOICE BOX II for 10 days, and if it isn't the finest value you've ever seen in a computer peripheral, the most challenging and provocative addition you've ever made to your system, return it in its original condition for a full refund.

THE ALIEN GROUP
27 West 23rd Street (212) 741-1770
New York, NY 10010

INCORPORATE THE SINGING HUMAN FACE
INTO YOUR PROGRAMS AND GAMES

The New VOICE BOX II for ATARI plugs into the serial port of the ATARI 400/800 with sound coming out of the TV/monitor. 48K DISK is required. It has all of the features of the original VOICE BOX plus many exciting new hardware and software features:

- The ability to sing with voice and 3 part music.
- A library of 30 famous songs.
- A comprehensive music system that allows the user to easily enter or modify new songs.
- Software that can convert the bottom two rows of the ATARI keyboard into a piano with a range of 3½ octaves using the shift and control keys.
- Programmable musical sound effects such as tremolo, vibrato, glissando and click track.
- A singing *human* face with lip-sync animation designed by Jerry White.
- A talking or singing ***ALIEN*** face with software that allows the user to change the face and 8 mouth patterns as he sees fit.
- The ability to speak with inflection and feeling.
- Can speak in a foreign language with correct foreign spelling as input.
- A talk and spell program by Ron Kramer. Users can program any vocabulary for this spelling game. In fact, this program can even speak in a foreign language like French, where the user must spell the correct word in English, or vice versa.
- **GREEN GOBLINS**–A talking arcade game by John Wilson.
- Random Sentence Generator–An amusing grammar game that helps teach school children to identify parts of speech and recognize a variety of sentence structures.
- **NUMBER SPEAK**–A subroutine by Scott Matthews that converts up to a 9 digit number into normal English pronunciation. Ideal for building your own math games.
- **STUD POKER**–A talking poker game by Jerry White.
- The screen never blanks out while talking or singing.
- Singing or speaking subroutines can be incorporated into your programs, requiring as little as 100 bytes of RAM plus 5 bytes for each word.
- Entries into the $5000 talking or singing game contest can be written using the VOICE BOX II–send for contest information.
- Price $169.00 includes VOICE BOX II and all of the above software.
- Inquire about our discounts for educational institutions.

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.

Atari is a registered trademark of Warner Communications.

ROM

VOLUME 1, ISSUE 2

FEATURES/ARTICLES

PROGRAMS

Columns

ROM STAFF

Editor/Publisher
Peter Ellison

Art Director
Paul Slevin

Technical Division
Bob Cockroft

Photographer
Jason Cockroft

Contributors:
Tim Reekie
Geoff Corry
Toung Tran
Ed Snyders

Domestic Subscriptions: 1 yr. 6 issues(bimonthly)$12;12 issues $22;18 issues $32. Send subscriptions orders to ROM P.O. BOX 252, Maple Ridge, B.C. V2X 7G1, TEL. (604) 462-9177 or 467-4658.

Program listings should be provided in printed form. Articles should be as typed copy in upper and lower case with double spacing. By submitting articles to ROM authors are able to collect royalties. All articles accepted for publication will become exclusive property of ROM. If the article is not accepted the only way it will be returned is, if a self addressed envelope is enclosed.

Editorial

Well we made it. The second issue of R.O.M. is out and we at ROM hope it will be a smashing sucess. As you can see we've got more color on the cover and hope to add some color to the inside in the issues to come.

It seems almost everyday that new third party software companies are starting up. The market is beginning to fill up, but there still seems to be room for all these new companies. With the flooding of the market it leaves more room for garbage software to creep in. We at ROM are going to try and show you what software is good and what software isn't worth the disk it is put on.

I remember about three years ago when most of the software available for the Atari was written in Basic. Then as the years passed, more and more machine language programmers came out of the woodwork. If the program was written in machine, it was a best seller, even if the program wasn't that good. Now with the many machine language programs on the market one must be careful on what he or she should buy.

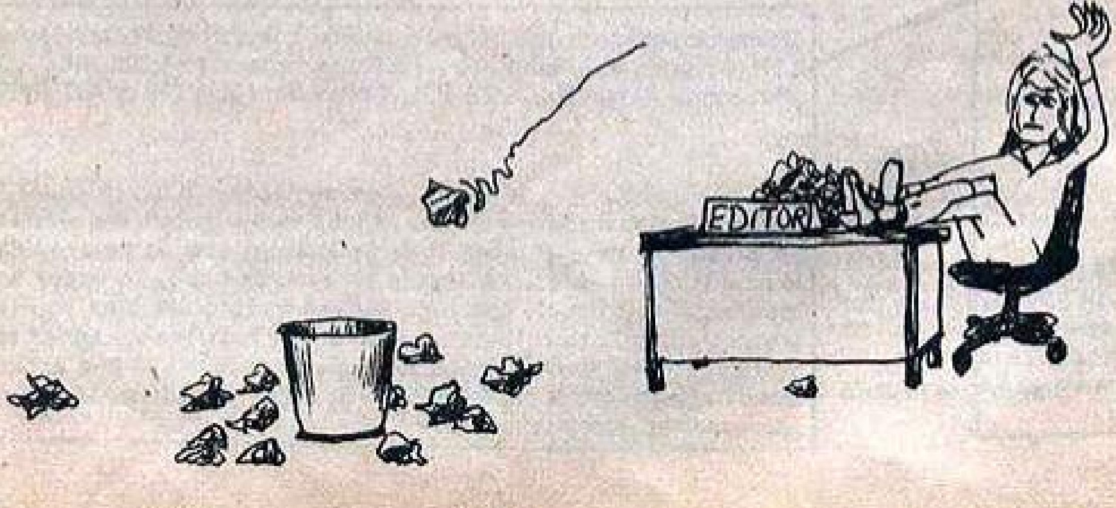
Each issue of ROM will contain a machine language game that has arcade quality graphics and sound. We know that there are many Basic programmers trying to get a grasp on what machine language is all about. We're going to give the assembly language listing for each game and a Basic listing for all of you that don't have an assembler and just want to play the game. The machine language game in this issue is called "Titan Lander" and is Lunar Lander Type game for up to four players. Each of our games will try and keep you entertained for hours.

Another thing that we are starting next issue is disk and cassete subscriptions of ROM. These subscriptions will contain all of the programs in ROM plus a copy of the magazine. We know how frustrating it is to type in a program from a magazine and it doesn't run because of a stupid typing mistake.

We're encouraging our readers to write us letters and tell us what they think of the magazine and what we could do to improve it. We plan after the third issue to have full color and gloss throughout but that is a little expensive for right now. Also if any of you can write programs in Basic, Assembler, Pascal, or Forth we'd be glad to publish them. We pay well and give bonuses for really excellent programs.

That seems to be all, so happy programming and so long for now.

P.S. Rumour has it that Alan Alda star of *M*A*S*H* will be doing something else besides being in old reruns. He will be the Bill Cosby of the Atari computer. He will be helping promote the Atari with television commercials, etc. Way da go ATARI!



Starting Page
Character Graphics - Part I
by Geoff Corry

Here is a way to put a nice picture on the screen that you can use in your own programs. This method uses text type characters that you have modified to build up the elements of your picture. This is Character Set Graphics.

As we go along I will give some references of some books and magazine articles that will explain things in more detail. The numbers refer to the list of publications at the bottom of the page.

First we will define some memory locations and set the computer into GR. 0 mode, otherwise some funny things may happen. See reference:-(5,7)

```
10 RAMTOP=106:CHBAS=756:CHORG=57344
20 GRAPHICS 0
```

Next we have to find a safe place to store our modified character set in memory. Here are some more references. (4,6,8,9;p55-56)

```
30 RAMNEW=PEEK(RAMTOP)-8
```

What we did here was to move the top of available memory down 8 pages (one page of memory is 256 memory locations). The character set that we are going to play with can now be copied into the safe area well above the available memory.

```
40 START=RAMNEW*256
```

'START' is the memory location where our relocated character set will start.

```
50 FOR CH=0 TO 1023
60 POKE START+CH,PEEK(CHORG+CH)
70 NEXT CH
```

Now we have finally copied the original character set (starting at 'CHORG') down into the new area. Each character takes 8 memory locations, giving a total of 1024 (8*128 char.) locations. This took time to do this in BASIC (approx. 15 sec.), so for those who want a bit more speed, see:- (9;p.85,86)

Now that the character set has been copied down into this new area, we have to change a signpost that still directs the computer to get it's characters at the old location.

```
80 POKE CHRBAS,RAMNEW
```

Once this line has been processed, all the characters will be obtained from the new location. 'SYSTEM RESET' or a 'POKE 756,224 will reactivate the original character set.

Below is part of the memory map to help visualize what has gone on here.

```
         ----------------------------------
         :             (CHORG+1023)58367:
4 pages   RESIDENT CHARICTER SET         :
         :57344(CHORG)                   :
         ----------------------------------
         :      F/P ROUTINES             :
         :       I/O CHIPS               :
         :       UNUSED AREA             :
         ----------------------------------
         :                          49151:
         :                               :
32 pages     BASIC CARTRIDGE             :
         :                               :
         :40960                          :
         ----------------------------------
         :                  :RAMTOP=160
         :                  :            :
         :                  (see below)
         :                  :            :
         :                  --------------
         :                  :            :
         :          ----------           :
         :          :RAMTOP=96:          :
         :          :         :          :
         :          (see below)          :
         :          :         :          :
         :          -----------          :
         :          :         : 48K MEM. :
         :          :         : EQUIP'D  :
         : -------- :         :          :
         :RAMTOP    : 32K MEM.:          :
         : =64      : EQUIP'D :          :
         :(see      :         :          :
         : below)   :         :          :
         -----------          :          :
         :          :         :          :
         :16K MEM   :         :          :
         :EQUIP'D   :         :          :
         :          :         :          :
         ----------------------------------
```

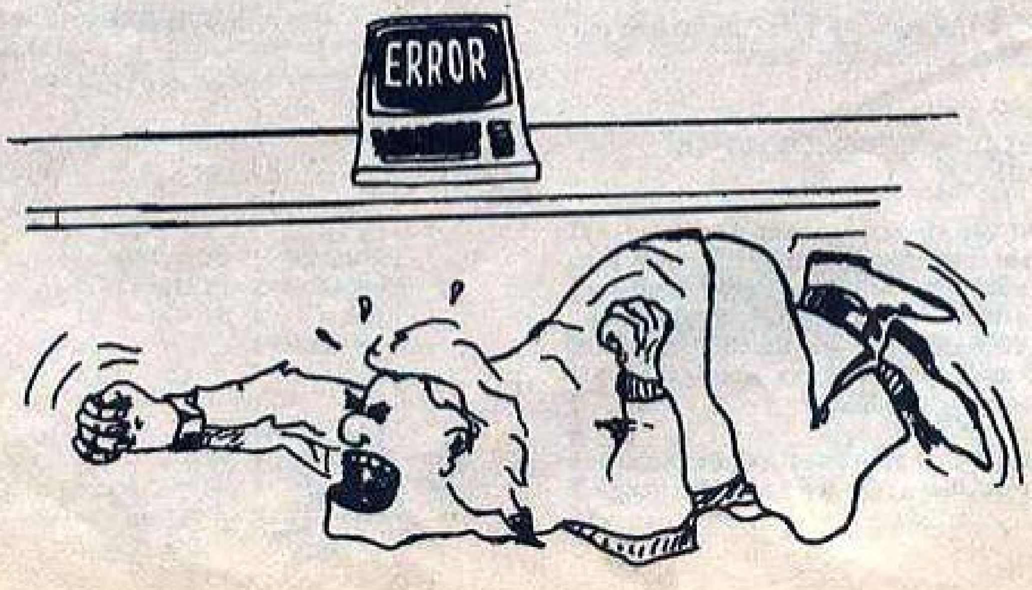
Starting Page cont'd.

You may have wondered why we had to move the character set down in the first place. Well, the original character set is 'cast in silicon' in the Atari factory and is Read Only Memory. We have moved it into an area called Random Access Memory where we can actually change any of the set as desired.

In the next issue, we will modify some of the characters and place them on the screen in a pattern to form a picture. In the meantime, type in these lines (10 to 80), and then 'RUN' the program. If you get some funny results, such as the wrong letters coming up when typing, just push 'SYSTEM RESET' and then list and check your program.

REFERENCES:-

1.ATARI 400/800 BASIC REFERENCE MANUAL.....................P. 55
2"........"...............P. 56
3"........"............APPEN. C
4"........"............APPEN. D
5"........"............APPEN. I
6.YOUR ATARI COMPUTER.P.291-29
7"........"............APPEN. F
8.DE RE ATARI......CHAP.3,P.4-8
9.COMPUTE'S FIRST BOOK OF ATARI GRAPHICS..............CHAP. 3



The Raving Reviewer

by Tim Reekie

FLOYD OF THE JUNGLE

MicroProse Software
One Caribou Court
Parkton, Maryland 21120
$29.95 Disk or Cassete; 32K

In the long list of Donkey-Kong like games, Atarians now have 'Floyd of the Jungle' by MicroProse Software. In this case however, Donkey Kong has been surpassed in at least two areas! 1-4 players can play simulataneously, and the background graphics are done exceptionally well.

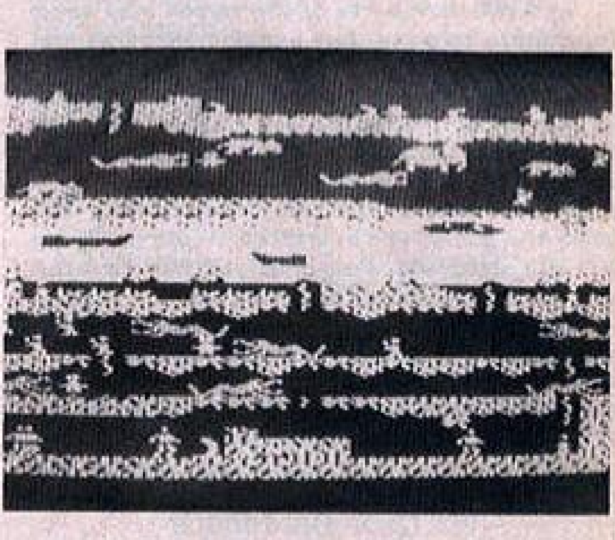
<u>The Game</u>

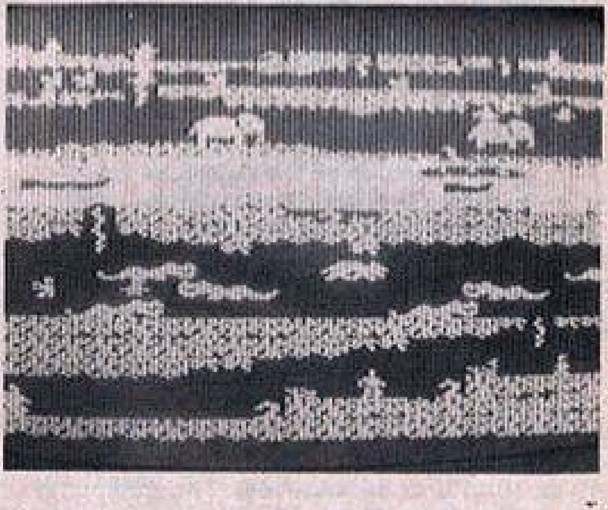
As I mentioned, Floyd of the Jungle can be played by 1-4 players simultaneously, with each Floyd capable of having a handicap(1 or 2 sore legs) for advance players. There are three difficulty levels: Easy, Medium, and Tough, and a playto value, that is set by the players, which automatically defaults to 20.

The object of the game is to rescue the fair maiden(that sounds familiar), Janice from nothing and no one in particular. To do so, our hero must jump snakes, ride alligators, elephants, and other miscellaneous animals, while avoiding getting eaten and or trampled by them. On two of the five different screens there is also a torrential river with canoes floating by for Floyd to jump on to.

And then, of course, are the Pygmies shooting poisoned darts at poor old Floyd. This is where the game gets comical. Floyd avoids the darts by jumping over them(push button) and can pick up two points by punching the Pygmy. Floyd gets his sustenance by catching birds(1 point each). The first Floyd to rescue Janice collects 4 points with additional points being added for traversing the course in less than 60 seconds and/or with a handicap.

Scores are displayed after each screen is completed, along with a table declaring the breakdown of points(birds, pygmies, rescues). The first person to earn the number of points selected receives a nice suprise.





<u>THE RATINGS</u>

Until the game is mastered, it is very frustrating, as every death returns your Floyd to the bottom of the screen. Even after the game is "mastered", the challenge is still very much present as you are now no longer racing

Raving Reviewer cont'd.

against the clock or trying to better your high scores, but your are now racing against someone who also knows the ropes(or in this case, the vines). One little mistake and...

At first some parts of the screen may seem impossible to pass, but experimentation and imagination will overcome these problems.

The graphics are incredible. So much seems to be crowded into such a small space. All over the screen, there is something happening. Before and after the game, the music is a superb three-part harmony, but during the game the sound is restricted to darts flying and Floyds dying. Personally, I had to stop and listen to discover that this was the case.

The instructions, besides being humorous, are quite informative, although for a while I was trying to punch the monkeys as they look similar to the pygmies. Experience sorts out these mistakes. This game has that lasting quality about it, with many innovative and downright funny ideas added for good measure.

FLOYD OF THE JUNGLE
Playability:9
Challenge:8.5
Graphics:10
Sound:9
Documentation:8
Overall Rating:9

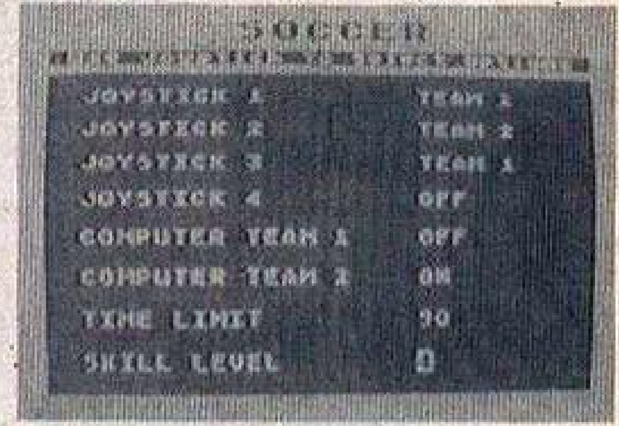
Soccer

Thorn EMI
1370 Avenue of the Americas
New York, New York. 10019

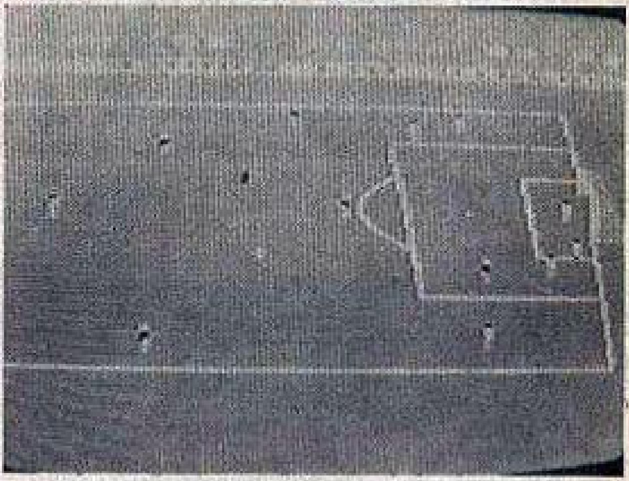
Hello soccer fans, and welcome to another fine night of soccer in Maple Ridge. We will be joining our on-the-spot reporter after these few words about the game.

GAME

SOCCER can be played from 1-4 players, two computer players, and every conceivable combination of these players including computer Vs. computer. The time limit can be set as 10, 45, or 90 minutes. The computer also has 1-4 skill levels which only the computer uses. And now to our on-the-spot reporter in Maple Ridge.



```
                 SOCCER
JOYSTICK 1              TEAM 1
JOYSTICK 2              TEAM 2
JOYSTICK 3              TEAM 1
JOYSTICK 4              OFF
COMPUTER TEAM 1         OFF
COMPUTER TEAM 2         ON
TIME LIMIT              90
SKILL LEVEL             D
```



ON THE SPOT

Good Evening, this is Coward Hosell, and welcome to my post game show.

Those two teams were really scrolling well out there today, although both seemed rather colorless. Those two players seemed to be all over the field(push button to become the nearest free player to ball). I came today expecting another of those unoriginal "sports" games, but instead found SOCCER to be a fast-paced challenging game. If you like Soccer, then you'll like SOCCER. From the turf at Thorn EMI stadium, this is Coward Hosell signing off.

Unfortunately, under this bed of roses, is

Continued on page 31

Display Lists
by Bob Cockroft

The display list is a program in your Atari computer that tells the Antic chip what and how to display information of the screen. One who has the ability to modify display lists has a much better understanding of graphics than one who does not. New graphic and text modes can be created or combined. Think of how your games could be improved. New and more interesting display lists could be used to enhance the graphics in a program.

Before I talk about display lists, you need to know how images are displayed on the screen. The television shoots an electron beam at the screen starting in the top left corner. This beam is moved horizontally until it reaches the right hand side of the screen where it moves down one vertical line. This process is repeated many times. These horizontal lines created by the beam are called scan lines. The Atari display has 192 of these scan lines located in the middle of the screen. By continually creating lower and lower horizontal lines the beam will eventually reach the bottom right corner where it is reset to the top left corner to enable a new screen to be drawn.

Before we have a look at the display list we need to find where it is. This can be easily accomplished by using the display list pointers.

```
ADDRESS=PEEK(560)+256*PEEK(561)
```

The variable 'ADDRESS' has the base (lowest) memory location of the display list. Now what we need is to have a peek at the entire display list(no pun intended). To do this you need to copy in the following program.

```
10 REM ***** PRINT OUT THE DISPLAY LIST
****
15 DIM DAT(176)
20 ? "THE DISPLY LIST OF WHICH GRAPHICS
MODE DO YOU WISH TO SEE?"
22 INPUT G
25 GRAPHICS G
30 DL=PEEK(560)+256*PEEK(561)
35 REM STORE DISPLY LIST IN AN ARRAY
40 FOR X=1 TO 176
52 DAT(X)=PEEK(X+DL-1)
50 NEXT X
55 REM PRINT ARRAY ON THE SCREEN
60 GRAPHICS 0
70 FOR X=1 TO 176
75 PRINT DAT(X);!? ",";
80 NEXT X
```

If you had correctly copied and ran the preceding program, there would be a list of numbers on the screen. This group of numbers is the display list. It is important to note that the long list of zeros at the end of some of the display lists, play no role. If you ran the program a number of times using different graphic modes, you would have probably noticed that the display list changes everytime the graphic mode changes. Although each graphic mode has its own unique display list, there is a common format you can follow. (It may be helpful to refer to table 1(an example display list) during the following discussion.

At the beginning of the list you will see 3 bytes which have the value '112'. These bytes create 24 blank scan lines at the top of the screen. The purpose of these is to move the display list to a readable location on the screen.

The next byte down the list, the LMS(the number in this location is in the area of 70). This is a 3 byte instruction which tells the Antic chip where to display the screen data.

The first byte tells the computer this is the LMS instruction and what graphic mode number to expect. The number generated in this byte is equal to the instruction register mode line number plus 64.

(the (IR) mode line number for graphics 2 is 7)

7+64=71

The (IR) mode line number is a number that the Antic uses to designate a particular graphics mode. This number does not correspond with the basic's graphic mode number. For example, the (IR) mode line number of graphic mode 8 is 15. Refer to table 2 for the (IR) mode line number for each graphic mode.

The second and third bytes are in the

Display Lists cont'd.

LSB/MSB form and give the address where the Antic will be jumped to in order to repeat or continue the list. By changing the values in these two locations, the screen can be scrolled in all directions. However, scrolling will not be discussed in this article because it has been already covered in a separate article in this same magazine(refer to magazine index).

The display list that is given to you in table 1, contains a line list of 7's. These 7's are the (IR) mode lines. Each byte you see containing a (IR) mode number represents one horizontal line of that graphic mode the number represents. The first (IR) mode number stands for the highest line of graphics on the screen. The following (IR) numbers represent horizontal lines of graphics which are progressively lower. As you remember each graphic mode has its own particular (IR) mode number. By replacing any or all the (IR) mode numbers with (IR) mode numbers of a different graphic mode, the horizontal line which is controlled by the altered number is changed to the new graphic mode. In other words by altering the (IR) mode numbers you can change any horizontal line on the screen to any graphic mode you like.

Immediatley after the row of (IR) mode numbers is another LMS instruction. The value in this byte varies with graphic modes the same way the previous LMS instruction did. In the example given to you in table 2, '64' is added to '2', the (IR) mode number for graphics 0,

Therefore 2+64=66

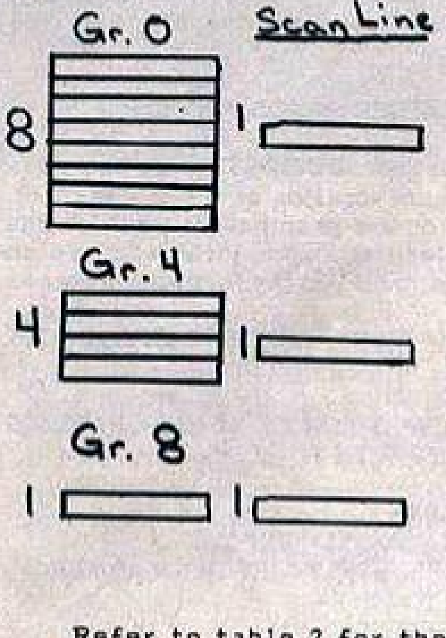
Like the previous LMS instruction the two following bytes represent the address where the Antic chip will jump to, to draw the screen.

The following group of 2's are the (IR) mode numbers for graphic mode 0. These numbers create the block of text graphics at the bottom of the screen.

The last instruction begins immediately after the 2's(note!in table 1 example this byte appears as a 65). This is another jump instruction. The first byte prepares the Antic for the jump and the last two locations give the destination. After the Antic goes through the display list, thus drawing the graphic mode, it is then reset by this instruction. In other words, these bytes force the Antic to draw the graphic mode repeatedly. Therefore, the second to last byte should equal PEEK(560) and the last byte PEEK(561), the starting address of the display list.

I have now gone through the format of the display list. It is time to use this knowledge in a practical application

Do you remember the scan lines? (Those lines created by the electron beam). The important thing is that there are 192 of them no matter which graphic mode you are in. With that in mind, it is time to understand how to use (IR) mode lines. Everytime you select a graphic mode, knowing it or not, you select a (IR) mode line. The higher the resolution of the graphic mode the larger the number of these (IR) mode lines. For example, graphic mode 8 has many times more (IR) mode lines than graphic mode 2] This is why graphics 8 is able to have higher resolution than graphics 2. Because graphics 2 and other low resolution graphics modes only use a few (IR) mode lines, each of these lines take up many scan lines. The diagram below displays the variations in size between different (IR) mode lines and scan lines!



Refer to table 2 for the numerical

Display Lists cont'd.

differences between (IR) mode lines and scan lines. It is important to remeber that the screen only has 192 scan lines and that the total number of scan lines used, cannot exceed this number. It would not take many mode line of a low resolution graphic mode to use all the available scan lines. For example, one (IR) mode line of graphic 2 would use 16 scan lines. Therefore the screen holds only 12 (IR) mode lines of graphic 2.

(Total scan lines=192: scan lines used for one (IR) line in graphics'2' =16)

Therefore 192/16=12 (IR) mode lines

Similiar calculations can be done for all the graphic modes. The important thing is that one must note the number of scan lines used for each (IR) mode line.

Enough of the theory; now it time for some programs you can use. Lets say, for example that you wished to create a screen which was half in graphics 3 and half in graphics 4. You would try to make the display list look like this.

The following is a graphic mode which is split between mode 3 and mode 4:

```
112,112,112,72,112,
158,8,8,8,8,8,8,8,8,8,
9,9,9,9,9,9,9,9,9,9,9,9,
9,9,9,9,9,9,9,9,
66,96,159,2,2,2,65,78,158
```

It is important to notice that I have initially set the graphic mode to the first byte on the screen. With this modified graphic mode the top of the screen will appear in graphics 3 and the bottom is graphics 4 (note that the rows of 8's create graphic mode 3 and the 9's create graphic mode 4.

In the final program I will attempt to combine everything I have said. It will also provide you with a workable example in which to develop your own programs from. There is only one small item in this program which I have not mentioned to you yet. As you probably know graphic modes fall into two major categories: text and no-text modes.

The text modes are graphics mode 0,1,and 2. The non-text modes are all the rest. The importance of this is that the computer must be told how you wish your information to be displayed on the screen; in text or graphics. For example, if you were to attempt to input graphics 8 (IR) modes lines into a graphics 0 screen without letting the Antic know to change to a non-text mode during the graphics 8 section, you would not be able to draw in this graphics 8 area. To let the computer know how to display information on the screen, poke in address 87 and the graphic mode you will be using. For example, if you were to use graphics 5 then POKE 87,5. This technique is only necessary when you are changing from a text mode to a no-text mode or the reverse.

```
1 REM ** MODIFIED DISPLAY LIST
PROGRAM 2 GRAPHICS 8:COLOR 1:OLOT
100,100:DRAWTO 120,100
3 DRAWTO 120,120:DRAWTO 100,120:DRAWTO
100,100
4 DIM B(200)
5 C1=0
6 B=PEEK(88)+256*PEEK(89)
10 GRAPHICS 2+16
12 COLOR 1
20 DL=PEEK(560)+256*PEEK(561)
25 REM GRAPHICS MODE 2
30 C=DL+6
40 POKE C,7
45 C=C+1
50 IF C1<4 THEN 40
55 REM GRAPHCIS MODE 8
60 C1=0
65 POKE C,15
70 C=C+1
72 C1=C1+1
75 IF C1<80 THEN 65
80 REM RESET LMS JUMP INSTRUCTION
100 POKE C,65
105 C=C+1
110 POKE C,PEEK(560)
115 C=C+1
117 POKE C,PEEK(561)
125 C2=0
128 REM STORE DL VALUES IN AN ARRAY
130 FOR L=DL TO C
135 C2=C2+1
```

Display Lists cont'd.

```
140 B(C2)=PEEK(LO
150 NEXT L
155 REM PRINT RESULT OUT ON THE SCREEN
160 GRAPHICS 0
170 FOR WT=1 TO C2
175 PRINT B(WT);
180 PRINT ",";
190 NEXT WT
```

Table 1
The Display List for graphics mode 2

112 This instruction will place 8 blank lines on the screen
112 8 blank lines
112 8 blank lines
71 First byte of the LMS (7+64)
112 This is the address of the first line of screen data
158 Therefore:112+256*158=address
7 The following 7's are the (IR) mode lines
7 Each (IR) mode line in this example uses 16 scan lines(see table 2)
7 Therefor there is 160 scan lines use
7
7
7
7
7
7
66 LMS instruction with graphic(0) (IR) number added
96 Starting address of the text window at the bottom of the screen
159 96+256*159=address
2 Text window
2
2
65 This 3 byte instruction returns the Antic to the top of the list
88 Address for the beginning of the Display List
158 88+256*158=address'

Table 2
Graphic mode statistics

Graphic(x)	Type	No. of scan lines per mode line
0	text	8
1	text	8
2	text	16
3	non-text	8
4	non-text	4
5	non-text	4
6	non-text	2
7	non-text	2
8	non-text	1

Graphic(x)	(IR) number	No. of mode lines
0	2	24
1	6	24
2	7	12
3	8	24
4	9	48
5	10	48
6	11	96
7	13	96
8	15	192

The Rating Scale

By Jake the Software Dude.

This table below is a reference sheet that can be referred to when reading the numbers given to each product review. It will help make clear why some of my numbers seem kind of low.

Rating Table

10-Totally original, outstanding.
9 -Excellent, show your girlfriend.
8 -Great, glad to have spent the money.
7 -Good, no complaints.
6 -Passable yet not to the programmer's potential.
5 -Passable, but dissapointing.
4 -Not worth the money.
3 -Don't show your friends.
2 -Don't show your mother.
1 -False advertising.
0 -Not recommended by the Surgeon General.

If there are any questions you can write to me in care of this magazine.

Jake The Software Dude

by Jason Cockroft

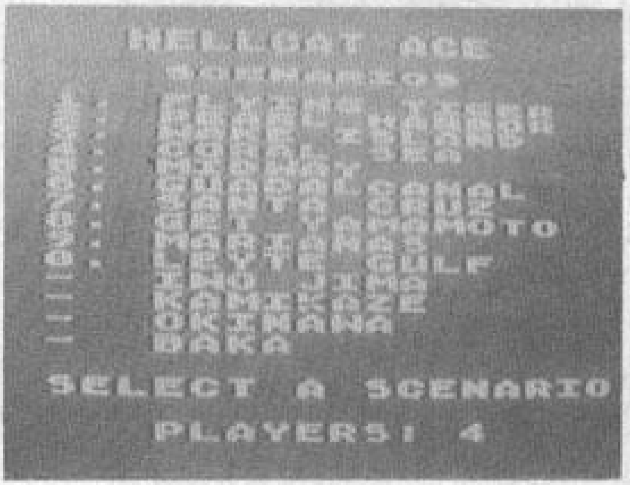
Aye readers, I'm Jake the Software Dude. I like to play games and play em ruff. My motto is, "if it's junk I'll give it the dump, if it's cool I'll let it rule." Some of the guys in my user gang say I have more braun than brains. So what if I go through a joystick every hour or if I have trouble using graphics mode 1. In the end I leave the computer smoking and the screen melting, with only the high score remaining!

In my first review, I'm going to review Hellcat Ace, a wild and rugged game created by Sid Meier. If you're the type of guy who stays up all night watching W.W.II fighter flicks, like I do, you'll go wild on this game.

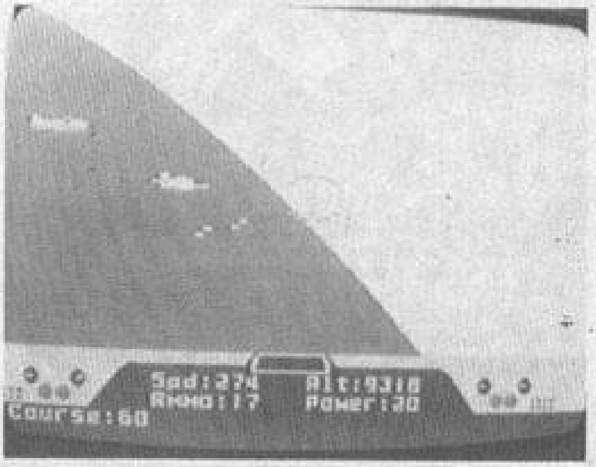
The basic layout of this game is based on you being a fighter pilot. You see what a W.W.II pilot in the Pacific would have seen! You see your power, ammo, heading, mirror, speed, and altitude gauges all on the console in front of you. But most impressive, which will burn your little eyeballs out, is the horizon. The reflection of the sun, and the changing color tones of the sky make the graphics, at least for the most parts, right on!

The basic format of the game allows the players to have a complete tour of the Pacific theater. You start as a Flying Tiger in August 1940 and hopefully finish with 5 kills or finish fighting in Okinawa in April 1945. In 14 different missions, you are asked to complete different tasks such as shoot down bombers, zeroes, scout planes, or avoid zeroes and even fight zeroes at night.

Gladly I find the game continually challenging and exciting. I find even after a month of continous practice,(I'm unemployed), I can only kill a zero 70% of the time and very rarely become an ace on skill level 3. Those intelligant zeroes dive, bank, weave and even climb into the sun. Sometimes they even challenge your skills by flying right down on the deck. The program justifiably makes the bombers and scout planes easier to shoot down than the zeroes. Meanwhile your own airplane can be damaged through his fire as you control your fighter with two jousticks; one of which you use to control your elevators and rudders and machine guns and the other to control your power and eject button. You can even ditch your own aircraft when it is nearly totally destroyed. For you chicken type of pilots there is another option of bailing out. Yet even here you can be killed if your picked up by the Japanese ship.



HELLCAT ACE
SCENARIOS
[illegible]
KAMIKAZE
OKINAWA
BAKA
SELECT A SCENARIO
PLAYERS: 4



Spd:274
Ammo:17
Alt:9318
Power:20
Course:60

Although the overall graphic presentation

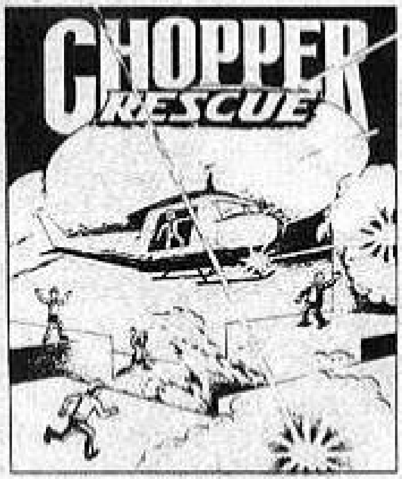
Continued on page 31

Experience the MicroProse Challenge ! ! !

Brief Blue 2, your computer wingman, on his mission and he will follow your instructions and lead into an air to air and air to ground battle against enemy territory . . . But, look out for Red Flight; they defend their territory and attack yours! . . . Scrolling, split screen challenge for 1-4 players . . . ATARI, 32K Disk or Cassette . . . $34.95.

Fly combat in a three-dimensional aerial dogfight over the Pacific during WWII . . . Requires real fighter pilot skills and maneuvers to survive to become an Ace or a W.G.F.P.J. . . . Try "SPITFIRE ACE" for aerial combat over Europe . . . 1-4 player combat . . . ATARI, 40K Disk or 32K Cassette . . . $29.95.

Race through the dangerous jungle avoiding the elephants, alligators, pigmies with poison darts, and other jungle perils to be the first to rescue the lovely Janice! . . . A zany footrace for 1-4 simultaneous players . . . ATARI, 32K Disk or Cassette . . . $29.95.



Pilot you nimble assault helicopter through the complex underground labyrinths. Destroy the enemy aircraft and defenses while you fight your way to rescue the captives trapped inside . . . Solo or team excitement for 1-8 players . . . ATARI, 32K Disk or Cassette . . . $29.95.

MicroProse Software is dedicated to bringing you games that challenge you far beyond the first few plays. We promise you hours of excitement and pleasure. Our games are created using "MicroProse", our own proprietary assembly language gaming system, and are available at select computer stores.

If you can not find our games at your computer store, you can order by MasterCard or Visa, money order, COD or check. Add $2.50 for postage and handling. MD residents add 5% sales tax. Call or write:

MicroProse Software 1 Caribou Court, Parkton, MD 21120, (301) 357-4739

DEALER INQUIRIES WELCOME!

ATARI is the registered trademark of ATARI, Inc.

Yellow Brick Road

by Peter Ellison

In the last issue of ROM I showed how one can use the keyboard for input by PEEKing into the Atari's memory. In this issue I'm going to show you how easy it is to use the Console keys (The three yellow keys at the right side of the terminal). These keys are very useful for games, business programs, or whatever different selections need to be made. To set up this program we must first PEEK into the memory at 53279 decimal. This checks to see what special purpose key is being pressed. Below is a table that shows if PEEK(53279) equals that number then that key is being pressed.

Table 1

7=no key pressed
6=START key pressed
5=SELECT key pressed
4=SELECT and START keys pressed
3=OPTION key pressed
2=OPTION and SELECT keys pressed
1=OPTION and SELECT keys pressed
0=OPTION, SELECT, and START pressed

The program below checks to see what key is being pressed and then prints out what is being pressed.

```
10 A=PEEK(53279)
20 ON A+1 GOTO 25,30,35,40,45,50,55,60
25 ? "SELECT, OPTION, START":GOTO 10
30 ? "OPTION, SELECT       ":GOTO 10
35 ? "OPTION, START        ":GOTO 10
40 ? "OPTION             ":GOTO 10
45 ? "SELECT, START        ":GOTO 10
50 ? "SELECT             ":GOTO 10
55 ? "START              ":GOTO 10
60 ? "NONE               ":GOTO 10
```

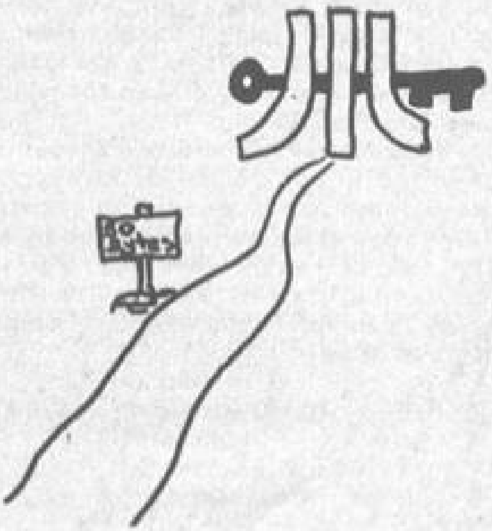
Remember next time you make a game or program that requires changing a level or number of players use the special purpose keys, their easy to use.

The second thing that will be discussed in this section is, "The Cursor". Yes, the little white thing that flashes on and off while you're typing in programs. This cursor can change color, blink, move, be positioned, and a whole lot more. Just by poking a number, each of these things can be done. For instance by typing POKE 752,1 you can turn the cursor completely off. Below is a Table that shows the different things that can be done with the cursor.

The Cursor

NAME	DECIMAL	DESCRIPTION
LMARGIN	82	Left Margin
RMARGIN	83	Right Margin
ROWCRS	84	Cursor Row
COLCRS	85	TAB
TXTROW	656	Cursor Row
TXTCOL	657	Cursor Column
CHACT	755	4=Vert, 2=Norm, 1=Blank
CRSINH	752	0=ON, 1=OFF
COLOR1	709	Color of Cursor

As you can see from the table above the cursor can be made to do just about anything by poking the right locations. The color of the cursor can be made lighter or darker by typing in POKE 709,X where X=0-255. Also by typing in POKE 755,4 all of the written text is displayed upside down. The cursor is a useful thing in programming when one knows how to use it. In the next issue of 'The Yellow Brick Road', player/missile graphics will be discussed with all of the locations needed to use them.



Game Reviews

Buried Bucks

Reviewed by Ed Snyders
ANALOG SOFTWARE
P.O. BOX 23
Worchester,MA 01603

If you like sensitive controls and fantasic explosions this game is for you.

You fly a helicopter, equipped with seventy-five rapid-fire bombs into an area where "the bucks" are buried underground. Your mission is to uncover the "bucks", pick them up, and bring them back to your home base. Don't let me deceive you, this game is harder than it looks. Buried Bucks is a game of speed and strategy. This is because as soon as you start blasting away at the ground a World War II bomber starts dropping dirt to replace the ground you have blown away.

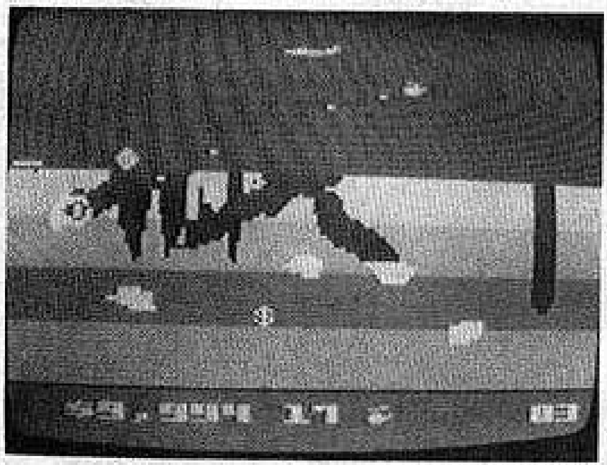
The first two levels can be finished by speed and guts but from the third level on up you need a plan of action. Luckily the chopper is very responsive and quick, but still if the flack from your own explosion doesn't get you the bomber may. One other thing to watch out for in the ground, is the water. If at all possible keep clear of it. More often then not my tunnels and holes seem to be filled when I'm firing around water.

When you reach the fifth level you are given a new nemesis. Missile lauchers that when land shoot fire balls at you. That along with all the dirt, flack and water make 'Buried Bucks' quite an experience. However the missile launchers can't land on uneven ground, so a quick spray of bombs back and forth across the screen should keep the majority of launchers from landing.

Remember you have to re-load in this game, the bullets don't last forever. This is usually when your holes start to get filled up. Your points which are represented by a dollar sign also decrease the longer you take to complete a level. This game is worth the money I paid for it and will keep me busy for hours to come.

Buried Bucks
Challenge-9
Graphics-8
Sound-8.5
Documentation-8

Overall Rating-8.3



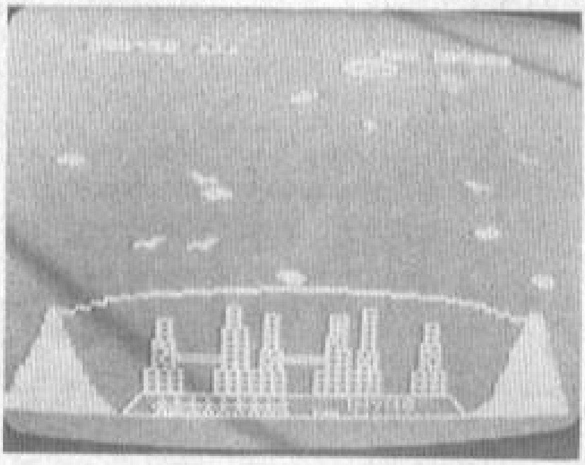
Meteor Storm

Reviewed by Peter Ellison
Royal Software
2160 W. 11th Ave.
Eugene,OR 97402
$29.95 Disk or Cassete;16K

You know this game is going to be great when the title screen appears and begins playing the theme from "Raiders of the Lost Ark" in four-part harmony. You can either listen to the whole song or press the start button to load in the game(Disk version).

A gigantic storm from space is approaching the planet Dracon at high speed. High above the planet's surface, a rescue ship waits helplessly, unable to penetrate the intense storm of meteor debris. You and your tiny fighter are the last hope for the people of Dracon. You must start from the docking bay of the rescue ship, manuever your way down to the city below, avoid meteors and their debris, and touch down safely on one of the three landing bridges. After picking up the survivor, you must clear a path back up to the rescue ship and carefully dock(Be careful that you don't hit the rescue ship!). A transport shuttle will then pick up your survivor and take them to safety. Your mission is completed when you have saved all of the people in the city.

Each city of Dracon has 10 survivors meaning to complete one level you must guide your tiny fighter down ten times. There is a shield over the city that is lowered every time your rescue ship nears the city. If you wait to long in getting out of the city with the survivor, you leave yourself vulnerable to a meteor hitting your city. If this happens 1000 points is deducted from your score.



Each piece of debris is a different color, making the game very pleasant to watch. The most notable thing about this game is the sound. Alot of games for the Atari have lousy sound because they have been translated from the Apple. This is not the case of this game. When leaving the rescue ship a rythmatic note begins to play. Then when landing in the city a little man makes a squeling noise as he waves his arms back and forth. The theme music from the movie "Superman" begins to play with the rythmatic note in the background.

When reaching the rescue ship with the survivor a nice little tune finishes the song off. The man is carried away and waves good-bye as he leaves.

There is a special feature called inviso that when the joystick is pulled back the ship dissapears making the debris pass through it. When leaving the city one can have the inviso on and continue to shoot lasers up and to both sides. The length of time the inviso can be held on increases after each survivor is rescued.

Meteor Storm is an excellent game but very difficult to master. It is said that if level ten is reached there is a special suprise(I can only reach level three). If you want a game that is very difficult, buy this.

METEOR STORM
Playability:8
Challenge:10
Graphics:8.2
Sound:9.5
Documentation:7
Overall Rating:8.1

The Battle of Shiloh

Stategic Simulations Inc.
465 Fairchild Drive, Suite 108,
Mountain View,CA 94043

"The Battle of Shiloh" by strategic simulations is a game that attempts to recreate this American civil war battle. The stronger Confederate's force objective is to destroy the union's army under General Grant before the north can reinforce and counter attack. Knowing that they have the advantage of surprise in the early stages of the game, the confederate player must force his way through the Union defenders and take the Pittsburg landing, a dock on the Tennessee River, to prevent the Union reinforcements from crossing.

This game provides a refreshing change from the typical themes of most wargames. Unlike many wargames, the Battle of Shiloh has an exciting and clear theme. The confederate player must take the Pittsburg landing and the Union player must prevent it. The army which is able to successfully obtain its goal greatly increases its changes of winning.

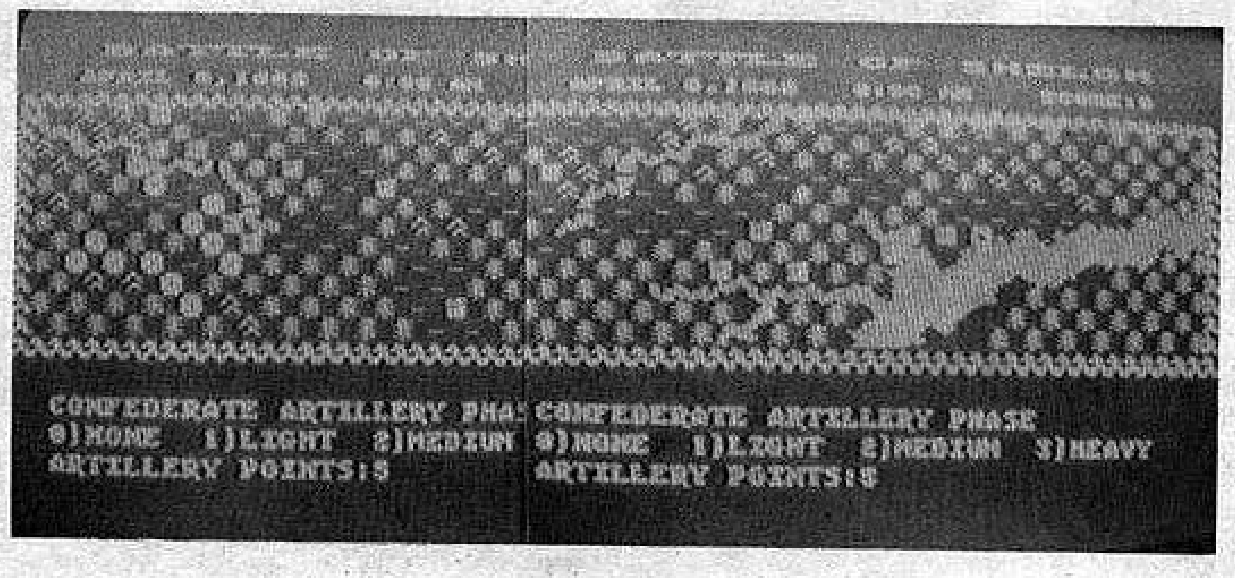
By allowing numerous attack strategies, this game provides a realistic approach to simulating this battle. The attacking side has four strategies to choose from ranging from all out attact to a more defensive recon. The defensive also has many strategies to choose from! ranging from an aggressive counter-attact to a more defensive mode of retreat. This program was created with both the daring and conservative player in mind.

One of the best aspects of this game is its flexiblity. One can be the union or the confederate playing against the computer or another player. The computer can even play against itself. In addition, there are multiple skill levels. Not only do the multiple skill setting make a provision for the varying abilities of players, but also increases the number of scenerios that can be created. An imbalance in military power can be created to produce different types of battle conditions.

This game unfortunately is a little slow. The whole process of moving the units seems to take to long. What causes this is units move individually and take to much time to make decisions. The problem is not as bad as I am making it sound. After all the computer is still much faster than any human player I have seen. And maybe the human player needs the time to think anyway.

The graphics in this game are not excellent, but good. The background depicts the countryside around and including the Tennessee River. This background is both properly colored and reasonablely detailed, showing hills, creeks, woods and fields. In addition, this battlefield scrolls horizontally. It is not a big battlefield, at least when compared with Eastern Front. However, this does not affect the game in a negative way. Being presented in the typical way, the units are presented as a square with an "x" in the center.

In conclusion, The Battle of Shiloh is a good war game. Its only real weakness would be the slowness of the computer's moving. It has good flexiblity and is quite entertaining.



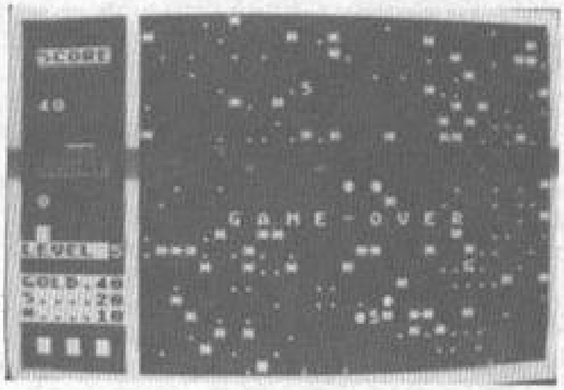
```
CONFEDERATE ARTILLERY PHA: CONFEDERATE ARTILLERY PHASE
0)NONE   1)LIGHT   2)MEDIUM 0)NONE   1)LIGHT   2)MEDIUM   3)HEAVY
ARTILLERY POINTS:5          ARTILLERY POINTS:5
```

Star Bomb

by Toung Tran

The year is 2122 and you as the last star pilot must destroy each of the stars by using your plasma cannon mounted on the front of your space craft. While you're trying to destroy the stars one to five Alien space ships (Depending on level) try to crash into you, thus destroying one of your three lives. Unfortunately, you are unable to destroy these space ships because they have a special cloaking device which makes it impossible to hit them. The only thing that can be done is to avoid them. There are also different types of ore and minerals that can be picked up for valuable points. They are D-Diamonds, P-Plutonium, G-Gold, and S-Silver. These will appear randomly on the screen as the game progresses.

The game uses the Atari's character set to draw the space ships. These ships are poked onto the screen and then moved around through poking new positions. The ship moves with the joystick and fires by pushing the red button. The game takes about a minute to intialize as it randomly plots the different stars. Remember don't touch the stars.



```
 10 DIM P$(3),A(5),X(5),Y(5),F(5),ROBOT(5)
:COLOR 2
 20 SCREEN=PEEK(88)+256*PEEK(89)
 30 GOTO 9000
 39 REM---CONTROL-SHIPS---
 40 S=STICK(0)
 42 IF S=15 THEN RETURN
 45 SOUND 0,150,10,10:POKE P,A(0):SOUND
0,0,0,0
 50 IF S=13 THEN M=1:SHIP=87:F=40:GOSUB
300
 60 IF S=14 THEN M=-1:SHIP=88:F=-40
:GOSUB 300
 70 IF S=7 THEN M=1:SHIP=65:F=1:GOSUB 300
 80 IF S=11 THEN M=-1:SHIP=68:F=-1:GOSUB
300
 85 P=SCREEN+X+40*Y:A(0)=PEEK(P)
 87 IF A(0)=84 THEN MAN=MAN-2:J=P:
GOSUB  5000:RETURN
 88 IF A(0)<>0 AND A(0)<>14 THEN GOSUB 800
 90 POKE P,SHIP:RETURN
 99 REM ---INITIALIZE---
 100 GOSUB 200
 110 H=SCREEN+A+40*B
 120 IF PEEK(H)=10 THEN FOR W=50 TO 100
 130 IF PEEK(H)=14 THEN SOUND
0,20,10,10:POKE H,10:SOUND 0,0,0,0
 140 IF INT(RND(0)*4)=1 AND PEEK(H)=0
THEN SOUND 0,100,10,10:POKE H,14:SOUND
0,0,0,0
 150 RETURN
 200 A=9+INT(RND(0)*30)
 210 B=1+INT(RND(0)*22):RETURN
 249 REM SET UP SHIPS
 250 V=INT(RND(0)*20):ON V GOTO 270
,280,280,290,290,290,291,291,291,291
 260 F(T)=10:RETURN
 270 F(T)=36:RETURN
 280 F(T)=48:RETURN
 290 F(T)=39:RETURN
 291 F(T)=51:RETURN
 299 REM CHECK SHIP
 300 IF PEEK(P+F)=128 THEN MAN=MAN-2:
J=P:GOSUB 5000:RETURN
 310 IF PEEK(P+F)=10 THEN MAN=MAN-2:
POKE P+F,0:J=P:GOSUB 5000:RETURN
 320 IF S=13 OR S=14 THEN Y=Y+M:RETURN
 330 IF S=11 OR S=7 THEN X=X+M:RETURN
 350 ? "↓(50)":POKE 85,15:? "G A M E - O V E R"
 355 IF STRIG(0)<>0 THEN 355
 360 ? "™DO YOU WANT TO PLAY AGAIN
(Y/N)";:INPUT P$
 365 IF SCORE>HI THEN HI=SCORE
 370 IF P$="Y" OR P$="YES" THEN 30
 380 IF P$="N" OR P$="NO" THEN END
 390 GOTO 360
 399 REM MOVE ALIEN SHIPS
 400 T=T2+INT(RND(0)*(10-T1)):IF T>5 THEN
```

Star Bomb cont'd.

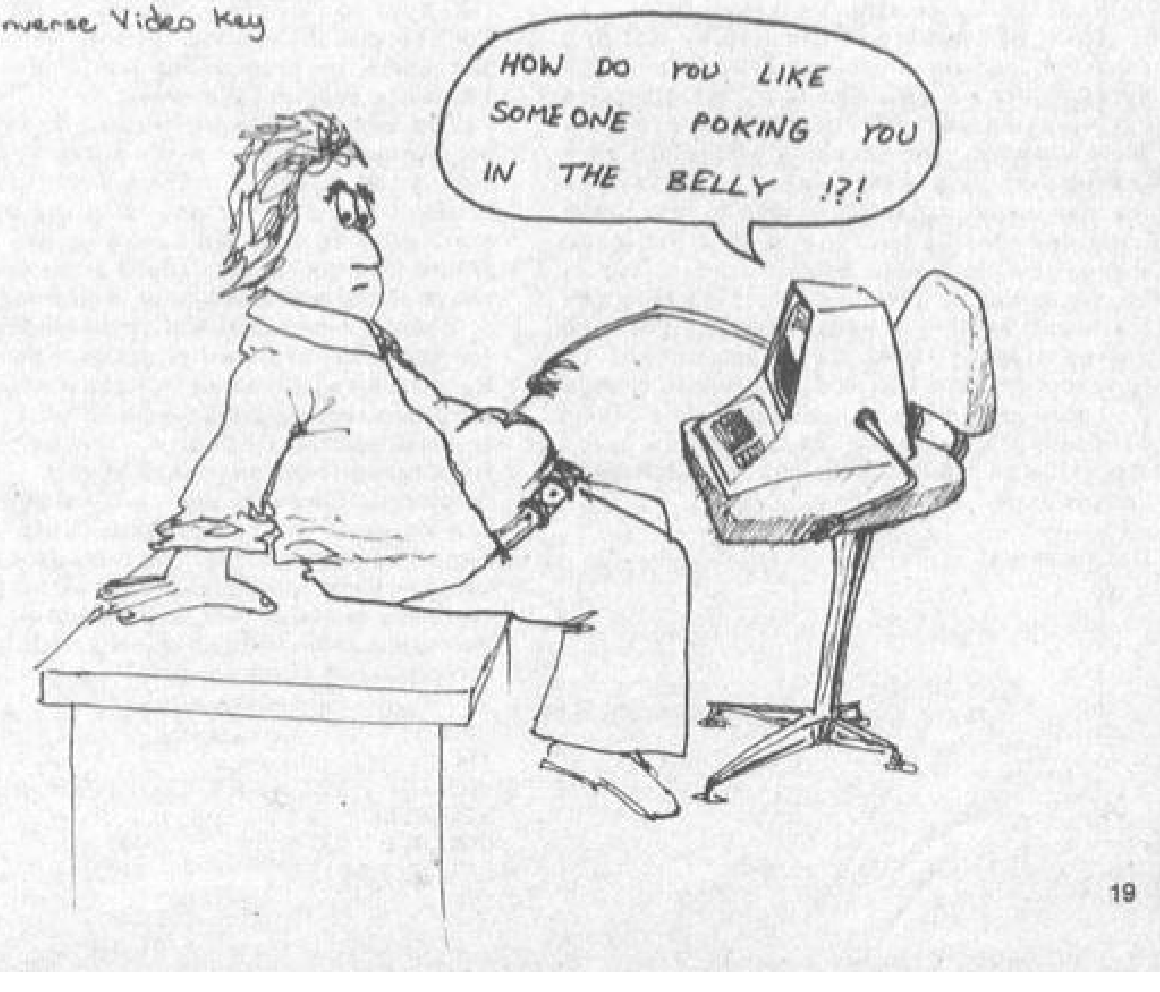
```
RETURN
 403 POKE ROBOT(T),F(T)
 405 IF Y(T)>Y THEN Y(T)=Y(T)-1
 410 IF Y(T)<Y THEN Y(T)=Y(T)+1
 420 IF X(T)>X THEN X(T)=X(T)-1
 430 IF X(T)<X THEN X(T)=X(T)+1
 435 IF X(T)<=8 OR X(T)>=39 OR Y(T)<=0 OR
Y(T)>=23 THEN 442
 440 ROBOT(T)=SCREEN+X(T)+40*Y(T)
:F(T)=PEEK(ROBOT(T))
 442 POKE ROBOT(T),84
 445 IF F(T)=65 OR F(T)=68 OR F(T)=87 OR
F(T)=88 THEN MAN=MAN-2:J=P:F(T)=0:GOSUB
5000:RETURN
 448 IF F(T)<>0 AND F(T)<>10 THEN GOSUB
6000
 460 RETURN
 469 REM KEEP SCORE
 470 SETCOLOR 2,16,1:FOR W=1 TO 23:POKE
85,0:? ;"▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨"!NEXT W:?
" ↑ (4 times) "
 500 ? "↑(8 times)":? "SCORE↓↓":? "↓↓↓→→HI↓
←(4 TIMES) SCORE":? :? HI
 501 ? :POKE 85,1:? ;"LEVEL ";LEVEL
 502 ? :POKE 85,1:? ;"D...100":POKE 85,1:?
;"P....50":POKE 85,1:? ;"G....40"
 503 POKE 85,1:? ;"S....20"
 504 POKE 85,1:? ;"*.....10":? "
":? "  ":? "    ↑"
 505 FOR W=1 TO 5 STEP 2:POKE SCREEN
+(1+W)+40*21,216:NEXT W
 509 SETCOLOR 2,16,1:FOR W=0 TO 39:POKE
SCREEN+W,128:NEXT W
 510 FOR W=0 TO 24:POKE SCREEN+40*W,
128:POKE SCREEN+8+40*W,128:POKE
SCREEN+40*W+39,128:NEXT W
 520 FOR W=0 TO 39:POKE SCREEN+W+40*23
,128:NEXT W
 600 FOR Y=1 TO 22:W1=9:W2=38:W3=1:IF
INT(Y/2)=Y/2 THEN W1=38:W2=9:W3=-1
 605 FOR X=W1 TO W2 STEP W3:PI=0:IF
INT(RND(0)*(60-LEVEL*10))=2 THEN PI=10
 610 IF INT(RND(0)*(30-LEVEL*5))=3 THEN
PI=14
 620 POKE SCREEN+X+40*Y,PI:NEXT X:NEXT
Y
 690 GOSUB 950:GOTO 9050
 800 SOUND 0,48,10,10:IF A(0)=36 THEN
SCORE=SCORE+100
 810 IF A(0)=48 THEN SCORE=SCORE+50
 820 IF A(0)=39 THEN SCORE=SCORE+40
 830 IF A(0)=51 THEN SCORE=SCORE+20
 840 ? " ";SCORE:A(0)=0:RETURN
 950 POKE SCREEN+(1+MAN)+40*21,128:
GOSUB 200:X=A:Y=B
 960 SHIP=88:P=SCREEN+X+40*Y:FOR W=100
TO 50 STEP -1:SOUND 0,W,10,10:POKE
P,216:POKE P,88:NEXT W:SOUND 0,0,0,0
 970 IF STICK(0)=15 THEN 970
 980 POKE P,0:RETURN
 3000 J=H:VER=40:HOR=1:O=0
 3005 GOSUB 6000
 3010 IF O=1 THEN RETURN
 3020 IF O=2 THEN J=P2:MAN=MAN-2:
GOSUB 5000 :A(0)=0:RETURN
 3080 GOTO 3005
 5000 FOR W=40 TO 70:SOUND 0,W,12,12
:NEXT W:SOUND 0,0,0,0:COUNT=0:VER=40
:HOR=1:O=0:A(0)=0
 5005 GOSUB 6000
 5010 IF O=1 THEN 5040
 5020 COUNT=COUNT+1:IF COUNT=3 THEN
5040
 5030 GOTO 5005
 5040 IF U=1 THEN RETURN
 5050 IF MAN<1 THEN 350
 5060 FOR Q=1 TO 200:NEXT Q
 5070 GOSUB 950:RETURN
 6000 POKE J,0
 6005 A(1)=PEEK(J+VER):A(2)=PEEK(J-VER
):A(3)=PEEK(J+HOR):A(4)=PEEK(J-HOR)
 6020 POKE J+VER,76:POKE J-VER,76
 6030 POKE J+HOR,76:POKE J-HOR,76
 6040 IF A(1)=128 OR A(2)=128 OR A(3)=128
OR A(4)=128 THEN O=1
 6041 IF A(1)=65 OR A(1)=68 OR A(1)=87 OR
A(1)=88 THEN O=2:P2=J+VER
 6042 IF A(2)=65 OR A(2)=68 OR A(2)=87 OR
A(2)=88 THEN O=2:P2=J-VER
 6043 IF A(3)=65 OR A(3)=68 OR A(3)=87 OR
A(3)=88 THEN O=2:P2=J+HOR
 6044 IF A(4)=65 OR A(4)=68 OR A(4)=87 OR
A(4)=88 THEN O=2:P2=J-HOR
 6050 POKE J+VER,A(1):POKE J-VER,A(2)
 6060 POKE J+HOR,A(3):POKE J-HOR,A(4)
 6070 VER=VER+40:HOR=HOR+1:RETURN
 7000 IF PEEK(P)=87 THEN N=40:D=93
 7010 IF PEEK(P)=88 THEN N=-40:D=92
 7020 IF PEEK(P)=65 THEN N=1:D=95
 7030 IF PEEK(P)=68 THEN N=-1:D=94
 7035 C=1:P1=P
 7040 A(5)=PEEK(P1+N*C):IF A(5)=128 OR
```

Star Bomb cont'd.

```
A(5)=84 THEN RETURN
  7045 IF.A(5)=10 THEN SCORE=SCORE+
10!J=P1+N*C!U=1!GOSUB 5000!U=0!? "↑
";SCORE:RETURN
  7060 POKE P1+N*C,D
  7066 IF C>2 THEN GOSUB 40:GOSUB 400
  7070 POKE P1+N*C,A(5)
  7080 C=C+1!GOTO 7040
  8000 REM EXAMINE ALIEN
  8010 IF F(T)=84 OR F(T)=92 OR F(T)=93 OR
F(T)=94 OR F(T)=95 OR F(T)=51 OR F(T)=39 OR
F(T)=36 OR F(T)=48 THEN F(T)=0
  8020 IF F(T)=14 THEN GOSUB 250
  8030 RETURN
  9000 REM ---MAIN-PROGRAM---
  9010 GRAPHICS 18!MAN=5!A(0)=0!
SCORE=0!LEVEL=1
  9012 POSITION 4,1!? #6;"*star
bomb*"!POSITION 4,10!? #6;"by toung tran"
#6;"by toung tran"
  9013 POSITION 6,4!? #6;"LEVEL"!POSITION
1,8!? #6;"PUSH START TO BEGIN"
```

▨ = Inverse Video Key

```
 .9014 POSITION 2,6!? #6;"SELECT FOR
LEVEL"
  9015 POSITION 11,4!? #6;LEVEL
  9016 L=PEEK(53279)!IF L=7 THEN 9016
  9017 IF L=5 THEN LEVEL=LEVEL+1!IF
LEVEL>5 THEN LEVEL=1
  9018 IF L=6 THEN 9020
  9019 GOTO 9015
  9020 T1=10-LEVEL !T2=6-LEVEL! GRAPHICS
0!SETCOLOR 2,16,1
  9030 IF LEVEL<=2 THEN T1=7-LEVEL
  9040 FOR E=1 TO 5!GOSUB 200!X(E)=A
!Y(E)=B!ROBOT(E)=SCREEN+X(E)+40*Y(E)
!F(E)=0!NEXT E!GOTO 470
  9050 GOSUB 40
  9070 IF STRIG(0)=0 THEN FOR W=10 TO
20!SOUND 0,W,10,10!NEXT W!SOUND 0,0,0,0
!GOSUB 7000
  9080 GOSUB 400!GOSUB 100
  9100 GOSUB 400
  9200 GOTO 9050
```



Display List Interrupts

by Bob Cockroft

Warning! It is recommended that reader should have a basic knowledge of display lists or have read the article in this magazine called 'DISPLAY LISTS' (see index).

Display list interrupts allows more colors to be presented on the screen. If you are a basic programmer, you have no doubt come across one of the main problems with graphics in basic, you cannot use more than four colors. But now with display list interrupts your problems are solved. By adding as many as 128 colors, your program displays can now be much more colorful.

As you already know, the display list is a program in the computer that is used by the Antic chip to display the screen. Although varying slightly with every graphic mode, the basic format remains constant. The display list's base address can be found by using location 560,561!the display list pointers

```
BASE=PEEK(560)+256*PEEK(561)
```

The first 3 bytes set the display list in a readable location on the screen. The next 3 bytes are the LMS. These bytes give the beginning address of where the screen data will be located. The following string of bytes are the ones we are interested in. Known as the instruction register(IR) mode bytes, these locations control the type of graphic mode that will be displayed. By each controlling an individual line of graphics, starting from the top to bottom, they are the largest single item on the display list. It is important to remember that the (IR) mode bytes both change in number and in the value contained in them with each graphic mode. The table below gives the value in the (IR) mode bytes with each graphic mode.

Table 1

Graphic mode	Value in (IR) mode byte
0	2
1	6
2	7
3	8
4	9
5	10
6	11
7	13
8	15

In this first article, in a series on display list interrupts, I will create a program using the Basic 'COLOR' command that will draw 3 colors on the screen. In addition, using display list interrupts, I will create one extra color by dividing the screen with color into 2 sections. Therefore I will have created 5 different colors on the screen; one more than the maximum for this graphic mode.

There are four steps to do to create a display list interrupt. First we want to determine where on the vertical plane we want to divide the screen with color. Remembering that each (IR) mode byte represents one line of graphics from top to bottom, one is able to find his desired dividing point by counting down the (IR) list the number of graphic lines before the point where you want the division. After this, all we need to do is add 128 to this (IR) byte to tell the computer to interrupt. For example, if the dividing point was to be in the middle, the programmer would need to add 128 to the middle (IR) byte.

The second thing we need to do is make a subroutine that tells the computer what to do during the interrupt. The subroutine I will create will be in machine language and will start at 1536(DEC). But before we are able to create this subroutine there are a few thing you need to know about color registers.

There are two different types of registers for color creation in the Atari computer; Hardware registers and shadow registers. Hardware registers are 'write only' locations. In other words, if a command was given to read these locations, only zero's would be outputted. Hardware registers are updated by the value in there corresponding shadow register everytime the Antic draws a screen. Unlike the hardware registers the shadow registers can be both read from and writen to. Below is a table of the hardware registers and corresponding shadow registers.

Table 2

COLOR	HARDWARE REGISTER	SHADOW REGISTER
COLOR 1	53271	709
COLOR 2	53272	710

COLOR 3	53273	711
BACKGROUND	53274	712

The following machine language subroutine will change the color of the background before the end of the screen by modifing the background hardware register before the shadow register can update it. In other words, we have changed the background color before the Antic has completed drawing the screen.

Machine Language Subroutine

```
Mem. Loc Value Assembly ;Comment
 1536    72   PHA ;PUSH 'A' ON THE STACK
 1537    169   LDA ;LOAD 'A'
 1538    1     #1 ;WITH ANY NO.
 1539    141   STA ;AVOID CHANGE
 1540    10   $0A ;IN MIDDLE OF
 1541    212   $D4 ;LINE
 1542    169   LDA ;LOAD 'A'
 1543    50   #50 ;WITH NEW COLOR
 1544    141   STA ;STORE NEW COL.
 1545    26   $1A ;IN HARDWARE
 1546    208   $D0 ;REGISTER
 1547    104   PLA ;REPLACE 'A'
 1548    64   RTS ;RETURN
```

The third thing we need to do is tell the computer where to jump after the interrupt. As you probably remember in step one we added 128 to one of the (IR) mode bytes of our choosing. Therefore forcing an interrupt when the Antic crossed this modified byte. Now we must tell the computer where to jump to in order to make the necessary changes in the color registers. The destination of our jump will be the machine language subroutine we created. Remembering that the beginning location of the subroutine is 1536 ($600 hex), we POKE this location into the address 512,513 dec. When an interrupt occurs, the computer looks at location 512,513 and jumps to that address they contain. It is important to note that the address stored in 512,513 is in LSB/MSB form. (Therefore 1536 decimal would be as '0' and '6'). (see below)

1536/256=6
Therefore:

POKE 512,0:POKE 513,6

The fourth and last step would be to enable a Non-maskable interrupt(NMI). This can simply be done by POKEing 54286 dec. with 192

The program below contains all the steps I have mentioned. It should help you in future programs of your own. It is important to note that only the bottom half of the screen is colored by Display List Interrupts. The blocks of color that will be seen have been created by the Basic's COLOR comand. By using a joystick, the number placed into the hardware register can be changed. Look at the bottom left corner of the screen for the value in the Hardware register.

```
2 REM **** DISPLAY LIST INTERRUPT ****
5 REM LOAD IN MACHINE SUBROUTINE
10 FOR X=1536 TO 1536+12
15 READ D
20 POKE X,D
30 NEXT X
35 REM LOAD LOCATION FOR THE
INTERRUPTS JUMP
40 POKE 512,0:POKE 513,6
50 GRAPHICS 5
60 DL=PEEK(560)+256*PEEK(561)
65 REM ADD 128 TO INTERRUPT BYTE
70 POKE DL+24,10+128
75 REM DRAW COLORED BLOCKS
80 COLOR 1
82 FOR C1=10 TO 20:FOR C1Y=10 TO 20
84 PLOT C1,C1Y:NEXT C1Y:NEXT C1
90 COLOR 2
92 FOR C2=40 TO 50:FOR C2Y=10 TO 20
94 PLOT C2,C2Y:NEXT C2Y:NEXT C2
100 COLOR 3
102 FOR C3=10 TO 20:FOR C3Y=30 TO 40
104 PLOT C3,C3Y:NEXT C3Y:NEXT C3
195 REM ENABLE (NMI)
200 POKE 54286,192
295 REM CHANGE VALUE PLACED IN
HARDWARE REGISTER
300 IF STICK(0)=11 AND P<255 THEN P=P+1
310 IF STICK(0)=7 AND P>0 THEN P=P-1
320 POKE 1543,P
330 PRINT P          500 DATA 72,169,1,141,10,212,169
340 GOTO 300         510 DATA 6,141,26,208,104,64
```

Books on the Shelf

by Peter Ellison

This is a new column in the magazine that will be devoted entirely to reviewing books made to enhance the Atari. Each issue I will review two or more books that are new to the market. Sometimes old books will be reviewed for those unfamiliar wtih them. In this very first article, three books from Elcomp will be reviewed . Remember if your company has any new Atari books on the market don't be afraid to send them in.

Games For The Atari

The cover of this book is very deceiving because it reads "Games for the Atari". A person who doesn't look inside might think its just a bunch of games, but if you turn to the front page of this book it then reads "or How to program your own games" which makes more sense. That is what this book is all about, "How to make games on the Atari."

Although it doesn't teach any machine language, it does have a machine language program in which you can type in. There are seven teaching sections which include "Player missile movement in machine language" and "Programming the Joystick". There are ten games that can be typed in and then played. These basic games vary from "Backgammon", which I liked the most, to "Gunfight", which must be type in with a machine language monitor.

Then at the back of the book is an Appendix that gives all the memory locations needed to use player missile graphics in basic. There is also a section called "Antic" which gives a little insight to how this microprocessor works. Character graphics are also discussed in a short section and so are display lists. This book contains everything needed to become an excellent game programmer in basic.

The documentation of the programs is good helping a new programmer understand why and how the program works. In my opinion $7.95 is a very low price for this very valuable book. This book is a must for anyone who has just bought an Atari computer.

How To Program Your Atari in 6502 Machinelanguage

This book is another fine book from Elcomp. This book helps to ease a basic programmer into the world of machine language. The author does this by first showing what the program would look like in basic and then gives the listing for it in machine language. For this book it is very important that you have either the Assembler Editor Cartridge from Atari or the ATAS-1 or ATMAS-1 from Elcomp. The book shows the differences of the three assemblers.

The book includes many useful machine language utilities such as a "Random Number Generator" required for many games like dice-games, mazesgames, etc. In the second chapter a programming model of 6502 CPU is given to help a Basic programmer begin to understand machine language. Another useful program is one that enables you to access the character set. The author takes for granted what you already know in basic and trys to use this to help you understand machine language.

This book isn't over the head of most people as many books on machine language are. This is why many people give up trying to learn how to program in machine language, but this book with some practice should help more people understand what goes on inside the Atari.

Books on the Shelf cont'd.

Forth on th ATARI
Learning by using

For the growing number of avid Atari owners buying Forth compilers, books like this one are a god-send. The cover is a little misleading as it shows a half-naked lady standing beside a man holding a rock with "Forth" carved on it. The art work is well done but may offend some people.

Enough about the cover. It isn't the cover most people buy books for. Inside the front cover is the table of contents ranging from chapter one(What is Forth?) to chapter nine(Appendix). The book starts out as if you have never used Forth before and gives a very thorough explanation of it. The book includes photographs and pictures to help enhance the reader's understanding of the material. The writing is also very clear as the author makes the topic easy to understand.

The thing about this book that makes it better then most Forth books is that it is directed specifically to the Atari and all of its special features. Chapter five is called "Sample Programs" where it gives several programs and definitions of words that are available in Forth. With this book being devoted to the Atari it gives very helpful programs such as using a graphic mode or programming a joystick routine. The programs are very helpful because no other Forth book would give them. If your reference manual that was received with your Forth didn't give them then you were out of luck.

This booklet besides being an excellent teacher of Forth gives a mailing list program, written in Forth, that can be used right after typing it in. The only thing wrong with this program is the printer has to be accessèd through the third joystick port by making up a little aparatus. I know this could be a bother for some.

I recommend this book to anyone who is interested in Forth and would like to get a head start on it. Elcomp has a free catalog that they will send out to anyone interested in their books or products. The address is given below:

Elcomp Publishing, Inc.
53 Redrock Lane
Pomona, CA 91766

Chart Maker

by Toung Tran

This program below is a graphing program that uses Atari's special character set to make the graph around the bars in the chart. The program first asks the user to input how long, or number of time. Next it asks how big you want the unit to be per block. This program can also be used by changing the letters to horses or numbers and make a gambling game.

```
10 DIM F(11),COUNT(11)
15 SCREEN=PEEK(88)+256*PEEK(89)
20 FOR W=0 TO 11:F(W)= SCREEN+6+W*3+
40*21:COUNT(W)=0:NEXT W
25 ? ") INPUT NUMBER OF TIME ";:INPUT
NUMBER
30 ? " INPUT UNIT PER BLOCK ";:INPUT
BLOCK:IF BLOCK<0 OR NUMBER<0 THEN 25
40 POKE 85,1:?
;":10QWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWE"
45 POKE 85,3:? ;"ASSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSD"
50 P=RND(0):FOR S=1 TO 9
60 ? 10-S;:POKE 85,3:? ;"ASSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSD"
70 POKE 85,3:? ;"ASSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSD"
80 NEXT S
90 POKE 85,1:?
;"0ZXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXC"
100 ?"' A B C D E F G H I J K"
105 FOR D=1 TO NUMBER
110 T=INT(RND(0)*11)
115 COUNT(T)=COUNT(T)+1
120 IF COUNT(T)>=BLOCK THEN
COUNT(T)=0:F(T)=F(T)-40:GOTO 300
150 NEXT D
200 POKE 764,255
210 IF PEEK(764)=255 THEN 210
220 POKE 764,255:GOTO 20
300 IF F(T)<SCREEN+40*2 THEN 150
310 POKE F(T),128:GOTO 150
```

A = CTRL A

Interview: Scott Adams

Interviewed by Peter Ellison

Adventure International has been in the software business for over five years now and has recently become very involved with the Atari market. They also make software that is available for the Trs-80, Apple, Texas Instruments, Northstar, and IBM. I think this wide selection of computers shows the versitility of the company. Many of the new programs for the Atari include Airline (a colorful game of high finance and corporate affairs), S.A.G.E.(Scott Adams' Graphic Editor), and The Disassembler (Compatible with most of the assemblers on the market).

Q. Scott, when did you first become interested in computers?

A. It was back in grade 3 at a science fair when I saw an IBM computer behind some glass. We were told not to go behind the glass. It was then that I wanted to be one of those people behind that glass.

Q. When did you start Adventure International?

A. I started Adventure International in 1978.

Q. How difficult was it to transfer your programs from the Apple to the Atari?

A. It was fairly easy to transfer programs from the Apple to the Atari as I have a Bachelor of Science in computers.

Q. Do you plan to use a voice synthesizer in any of your new adventure games?

A. Well we're supporting for the Atari and the Apple, the Blowtrack sythesizer and similar ones like it.

Q. Did you begin working out of an office or out of your home?

A. I started in a small bedroom which I used for my office.

Q. How big is your office at this time?

A. Over 11,000 square feet.

Q. Where do you get all of your ideas for an adventure?

A. Same way novelists or artists get their ideas, they just come.

Q. What type of theme do you like to dwell on best?

A. I don't have any preferences, I just enjoy writing adventures. At heart a frustrated writer, always wanting to write books. A tremendous avid reader I have thousands of books in my personal library and adventures have turned out to be my creative outlet.

Q. How long on the average does it take you to write one of your adventure games?

A. A couple of months.

Q. Now that Adventure International has grown so greatly in the last year, do you have time to oversee all your new software programmers?

A. I try to for sure.

Q. How many programmers do you have working for you at this time?

A. We're basically a publishing house for freelance authors and we have over 60 authors worldwide.

Q. What do you do when you are not programming?

A. Well running the business, in itself, is a full time job. I've got two kids that I spend alot of time with. I don't seem to have alot of free time. I'm either programming at the office or playing with the kids.

Q. What kind of deal do you offer to an outside programmer, such as Russ Wetmore, author of Preppie?

A. A very attractive one. We have a standard author pack we're glad to send to anyone that is interested in becoming a software programmer.

Q. What age group are your programs directed to?

A. From preschoolers on up. We try to find something for everybody.

Q. Do you ever use suggestions from letters and incorporate them into your own programs?

A. Usually not although somebody will come along and find a bug in a program and suggest a modification.

Q. Do you plan to make use of the new Atari 1200XL?

A. Right now we haven't, but you never know what the future will bring. Our biggest thing right now is making sure our current programs are compatible with the 1200. There are problems with that and we have just about got that licked. One main reason we haven't brought out programs specifically for the new 1200 is because so far its not selling as well as the 400.

Continued on page 31

SUBSCRIBE TO ROM
AND DON'T MISS AN ISSUE

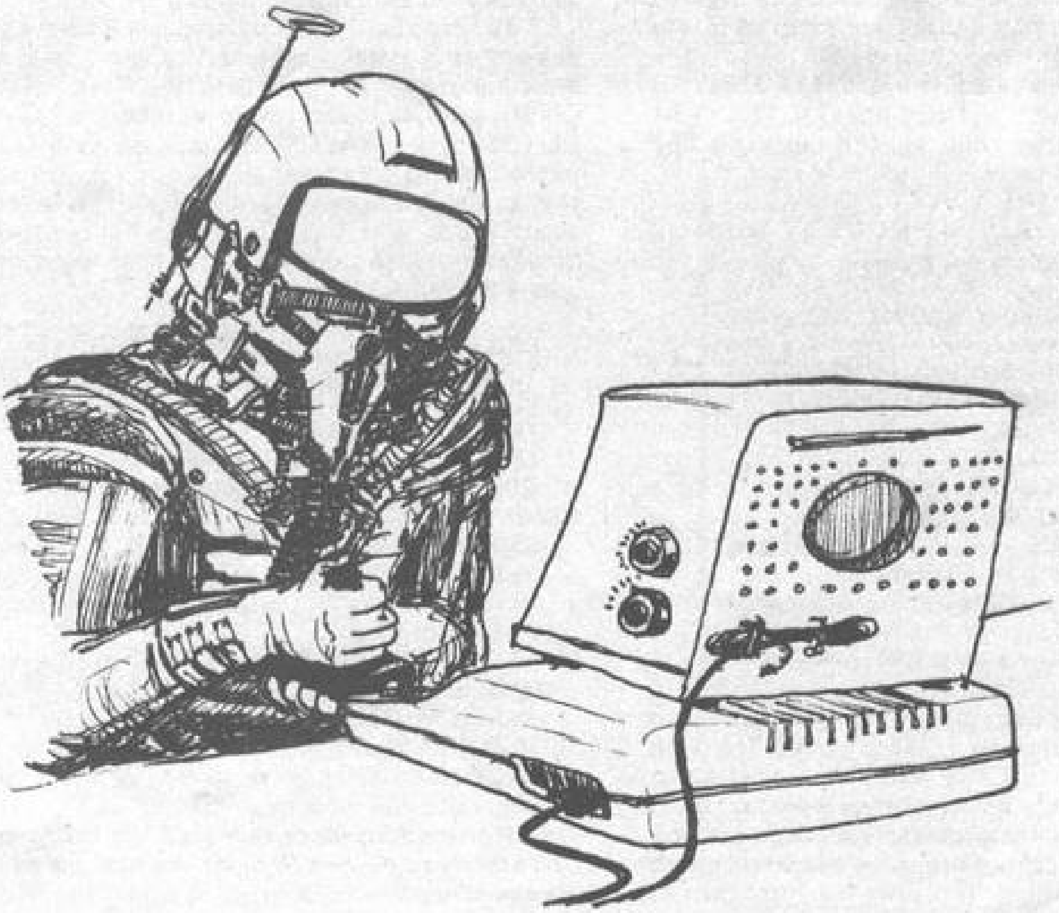
Mail To!
ROM
P.O. BOX 252,
Maple Ridge,B.C.
V2X 7G1

__ 1-Year(6 issues)$12.00
__ 2-Years(12 issues)$22.00
__ 3-Years(18 issues)$32.00
__ 1-Year Magazine and Cassette $30.00
__ 1-Year Magazine and Disk $50.00

__ Check enclosed. __ MASTER CARD __ VISA

Card No.______________________________ Exp. Date__________

Name____
Address____
City/Prov/Pos____



Scrolling Your Atari

by Bob Cockroft

In order to allow the screen to scroll horizontally one must find the starting address of the display list. This address is located in memory location 560,561 (230,231 hex)(symbol:SDLSTL). It is important to note that the starting address of the display list is using the LSB/MSB system. What this means is that the 560 represents the least significant bit of the display list address and the 561 the most significant bite. Therefore to get the correct address you must multiple the most significant bite by 256 and add to this the least significant bite.(see below)

```
ADDRESS=PEEK(560)+256*PEEK(561)
```

What we have now is only the starting address of the display list. What we need now is the rest of the value composing the display list. The remaining values are located in the higher locations immediately following the starting address of the display list. A complete picture of the display list can be easily found by copying and running the following program

```
    REM **** PRESENT DISPLAY LIST ON
SCREEN (GR.2) ***

    10 DIM DL(200)
    20 FINISH =176
    30 GRAPHICS 2
    40 DIS=PEEK(560)+256*PEEK(561)
    50 COUNT =1
    60 DL(COUNT)=PEEK(DIS+COUNT-1)
    70 COUNT=COUNT+1
    80 IF COUNT<=FINISH THEN 60
    90 GRAPHICS 0
    100 FOR C2=1 TO FINISH
    110 PRINT DL(C2);",";
    120 NEXT C2
    140 END
```

The display list values vary both in length and content with the graphic mode. Its main purpose is to tell the ANTIC chip what to display on the screen. But in this article, however, we will only concentrate on locations relative to scrolling. We will first attempt to understand the principles of horizontal scrolling then later move onto vertical scrolling

In order for us to begin to create a program that will scroll horizontally we must first locate the LMS instruction. The LMS is an 3 byte instruction that tells the ANTIC chip which address to jump to in order to repeat or continue the display list. This instruction is located 3 bytes after the display list base address. If you look at the top of the list given to by the above program you will notice that 3 has a value of 112 and then a value of 71. This 71 is the first byte of the LMS(note: this value 71 will change slightly with different graphics modes). The first byte of this instruction prepares the computer to jump. The second and third bytes is the address that it will jump to(note: this address is in LSB/MSB form).

Horizontally scrolling is done by incrementing or decrementing the value in the least significant byte of the LMS. This least significant byte is located immediately AFTER the first byte of the LMS and appears as a 112 in the list. At this point it is best for you to copy in the following program(note: the least significant byte being used to horizontal scroll is equal to the display list base address plus 4 (DL+4)).

```
    REM **** HORIZONTAL SCROLLING ****
    10 GR.7
    12 COLOR 1
    14 SETCOLOR 2,16,1
    15 REM ** DRAW FIGURE **
    20 PLOT 40,20:DRAWTO 40,60:DRAWTO
65,60
    25 DRAWTO 65,20:DRAWTO 40,20
    30 DL=PEEK(560)+256*PEEK(561)
    40 LSB=DL+4
    45 B=PEEK(LSB)
    50 IF STICK(0)=11 THEN B=B+1
    60 IF STICK(0)=7 THEN B=B-1
    79 REM ** POKE NEW VALUES**
    70 POKE LSB,B
    80 GOTO 50
```

If the above program was copied out correctly you now have seen the block move across the screen. careful observation will reveal if one moves the block in one direction

Scrolling Your Atari cont'd.

for a long period, the figure slowly moves up or down. This effect is a result of the fact that the screen is stored as one long row of bytes starting from the top left corner and moves horizontally in line until it reaches the lower right hand corner. Therefore when one changes the least significant byte by 40 the screen is scrolled vertically by one byte(note! this is a 40 character screen). You have just discovered vertical scrolling! Before you start celebrating it is best I tell you about a few complications I have not yet mentioned. It is true by just incrementing or decrementing the least signficant byte of the LMS you can scroll vertically, but not very far. This limitation became very obvious with only a little thought. After all, when you must add 40 every time you want to move vertically to a byte that can only hold a value between 0 and 255, it seems obvious you are not moving very far. However, there is a way to solve this. Remember the LMS's MOST SIGNIFICANT BYTE, the one following the least significant byte,(note! the value on the display list program was 158) by changing this we will end the problems. Every time the Least Significant Byte goes below 0 and 256 to the LSB to reset it and subtract 1 from the Most Signifcant byte. By doing this, the LMS pointer is increased by 256 bytes.

You must also subtract 256 from the LSB and add 1 to the MSB ever time the LSB goes above 255. After all this is done the new MSB and LSB value must be poked into their respective locations. If you are still confused or perhaps want to practice what you have learned, copy in the following program. It is designed to scroll both vertically and horizontally.

```
10 REM **VERT & HOR SCROLLING **
12 GR. 6
14 COLOR 1
15 SE 2,16,1
18 REM *DRAW FIGURE
20 PLOT 40,20:DRAWTO 40,60:DRAWTO 65,60
22 DRAWTO 65,20:DRAWTO 40,20
30 DL=PEEK(560)+256*PEEK(561)
40 LSB=DL+4
50 MSB=DL+5
60 BL=PEEK(LSB)
70 BM=PEEK(MSB)
79 REM *CHANGE VALUES*
80 IF STICK(0)=14 THEN BL=BL+40
85 IF STICK(0)=13 THEN BL=BL-40
90 IF STICK(0)=11 THEN BL=BL+1
95 IF STICK(0)=7 THEN BL=BL-1
100 IF BL<0 THEN BM=BM-1:BL=BL+256
110 IF BL>255 THEN BM=BM+1:BL=BL-256
180 IF BM<0 OR BM>255 THEN 80
199 REM *POKE NEW VALUES
200 POKE LSB,BL
210 POKE MSB,BM
220 GOTO 80
```

After running this program you may have noticed ghost images when moving vertically. Nothing can be done about this in basic. They result from the slowness of the basic and can only be eliminated by a faster language.

This article has not revealed all that can be revealed about scrolling, nor was it intended to. What I have tried to do is provided a simple and hopefully practical introduction to this useful technique. HAPPY SCROLLING!

Adventure Games - Part II

by Peter Ellison

Last issue 'Black Knight' was set up just by the designing of the castle on paper. In this issue brave Sir Lancelot will be created using player missile graphics. Also some creatures such as a dragon and a giant will also be made.

Lancelot since he is wearing Chain-mail armour will only be able to move eight spaces per move. This will be done in the final installment by having a machine language subroutine check the joystick each time and turn it off after one space has been made. This is very similar to the movement of Ali Baba(TM) which is an excellent system for any adventure. Last issue I said that was going to show you how to make the castle using data statements, instead I'm going to show a few of the characters that will be used in the final program. I decided to do it this way so that the entire program would be in the third and final installment. This is so a person wouldn't be forced to buy back copies of the magazine.

The first character is one of a dragon. The listing below will draw a yellow dragon on the screen by using four players to do this. Coming from his mouth is a red tongue. By using the four players, larger characters can be created, and as you will see in the next issue, moved. Type in the program listing below.

Dragon

```
45 REM SET UP PLAYER/MISSILE BASE
50 I=PEEK(106)-8:POKE 54279,I
55 REM PLAYERS 0 AND 1
60 J=I*256+516:K=I*256+644
65 REM PLAYERS 2 AND 3
70 L=I*256+768:M=I*256+896
95 GRAPHICS 2
100 SETCOLOR 2,11,6
105 COLOR 3
200 POKE 559,46
209 REM PLAYER 0
210 POKE J+1,0:POKE J+2,0
220 POKE J+3,1:POKE J+4,128:POKE J+5,252:POKE J+6,128
229 REM PLAYER 1
230 POKE K+1,1:POKE K+2,3:POKE K+3,2:POKE K+4,63:POKE K+5,3:POKE K+6,63
239 REM PLAYER 2
240 POKE L+4,128:POKE L+5,128:POKE L+6,128:POKE L+7,128:POKE L+8,131:POKE L+9,207
245 POKE L+10,255:POKE L+11,127:POKE L+12,63:POKE L+13,31:POKE L+14,12:POKE L+15,60:POKE L+16,240
249 REM PLAYER 3
250 POKE M+7,192:POKE M+8,240:POKE M+9,240:POKE M+10,248:POKE M+11,248:POKE M+12,252:POKE M+13,240:POKE M+14,60
255 POKE M+15,12:POKE M+16,15
475 REM DRAGON'S COLOR
480 POKE 704,50:POKE 705,41:POKE 706,43:POKE 707,43
483 REM DRAGON POSITION
485 DRAGON=100
490 DPOS=DRAGON
510 POKE 53277,3
1229 REM POKE DRAGON TO SCREEN
1230 POKE 53248,DPOS:POKE 53249,DPOS:POKE 53250,DPOS+8:POKE 53251,DPOS+16
```

Since only one monster will be moving at a time it is easy to use four players for one monster. Below is the character data for four other creatures, one of them being poor Lancelot. By deleting lines 209-255 from the above program just the few lines can be added to show the images of the other characters in the game. Below is Lancelot.

Lancelot

```
210 POKE J+1,56:POKE J+2,68
220 POKE J+3,130:POKE J+4,130:POKE J+5,84:POKE J+6,56:POKE J+7,16:POKE J+8,56:POKE J+9,16:POKE J+10,16
225 POKE J+11,40:POKE J+12,68
230 POKE K+1,0:POKE K+2,0:POKE K+3,40:POKE K+4,0:POKE K+5,16:POKE K+6,0:POKE K+7,0:POKE K+8,0
```

The next monster is one of a giant. This giant has a club in his hand in order to add to the graphics. To have the giant's club in the right position POKE 53250,DPOS+8 must be changed to POKE 53250,DPOS-8.

Giant

```
209 REM PLAYER 0
210 POKE J+1,62:POKE J+2,42
220 POKE J+3,54:POKE J+4,62:POKE J+5,8:POKE J+6,62:POKE J+7,28:POKE J+8,28:POKE J+9,128:POKE J+10,28:POKE J+11,28
```

Adventure Games - Part II cont'd.

```
 225 POKE J+12,20:POKE J+13,20:POKE
J+14,20:POKE J+15,20:POKE J+16,54
 229 REM PLAYER 1
 230 POKE K+2,20:POKE K+3,8:POKE
K+6,65:POKE K+7,193:POKE K+8,194:POKE
K+9,252:POKE K+10,64:POKE K+11,64
 235 POKE K+12,64
 239 REM PLAYER 2
 240 POKE L+8,6:POKE L+9,7:POKE
L+10,3:POKE L+11,1
```

The last two are that of a wolf and a mummy.

Wolf

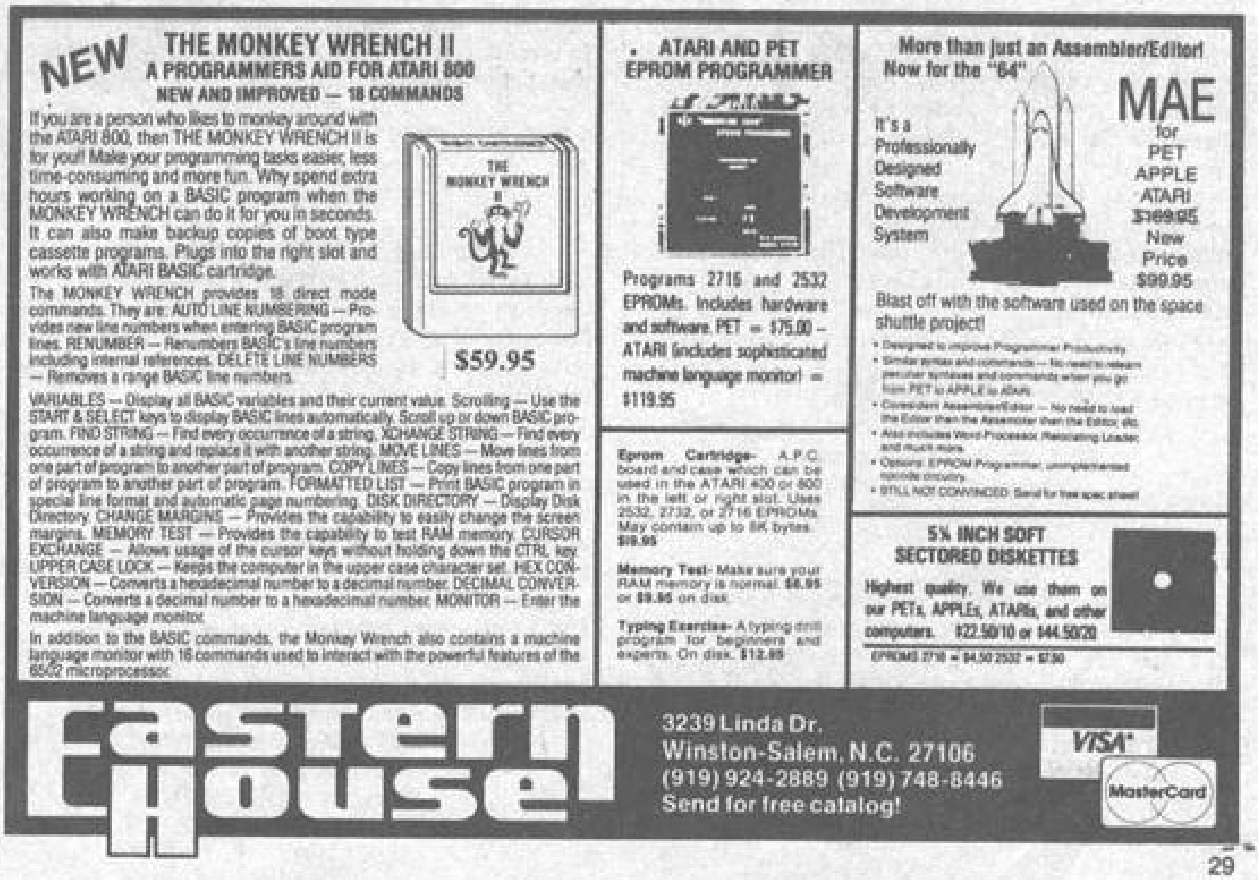
```
 REM PLAYER 0
 210 POKE J+1,32:POKE J+2,224:POKE
J+3,32:POKE J+4,32:POKE J+5,63:POKE
J+6,63:POKE J+7,63:POKE J+8,33
 220 POKE J+9,33:POKE J+10,33
```

Mummy

```
 209 REM PLAYER 0
 210 POKE J+1,28:POKE J+2,42:POKE
J+3,28:POKE J+5,127:POKE J+7,93
 220 POKE J+9,28:POKE J+11,20:POKE
J+13,20
```

```
 229 REM PLAYER 1
 230 POKE K+2,20:POKE K+4,8:POKE
K+6,93:POKE K+8,93:POKE K+10,20:POKE
K+12,20:POKE K+14,54
```

In closing remember POKE 704 through POKE 707 can be changed to give Lancelot or the monsters a different color. Next issue will be the entire listing of the game 'Black Knight', and believe me, it will be worth the wait. Until then, Happy Adventuring!



New Products

Two New Games From PDI

The first being Android, a game designed to challenge both beginning and advance game addicts. A remote-control android has been sent to a distant asteroid. The android's mission is to gather treasures from the labyrinth buried beneath the asteroid's surface. The labyrinth is guarded by armed robots. It has five levels, and the android must pass through all the levels to escape and return to earth with the treasures. You see the maze through the eyes of the android in 3-D. It requires 40K with an Atari 810 Disk Drive; an Atari 410 Program Recorder is needed to use optional voice track.

The second game called Clipper: Around The Horn In 1850 also includes an optional cassete voice track. As captain of one of the famed "greyhounds of the sea", the player must make his or her way from New York to San Francisco, via Cape Horn at the southernmost tip to South America. Shrewd planning and wise choices are needed to make this trip in the shortest possible time, with the greatest possible amount of money remaining in the treasury at voyage's end. There are many perils along the way, including fierce storms, deadly calms, and mutinies. Masts may be snapped by howling winds, illnesses my befall the crew, icebergs and submerged reefs may destroy the ship. The game features exceptional nautical graphics. The cassete version requires 24K. The disk version requires 32K. The two versions are sold together, in one package, for a retail price of $29.95.

Program Design Inc.
95 East Putnam Avenue
Greenwich, CT 06830
(203) 661-8799

A.E.

This new game from Broderbund Software puts you against squadrons of menacing sting rays that are streaking down from the sky to attack you. You're doomed to be pestered forever unless you drive these waves of A.E.("rays") out of the solar system, deep into the outer wastelands of space. A.E. is a fast-paced arcade style game with excellent graphics and eight levels of play. Available on disk with 48K. Played with joystick or paddle. Suggested retail price is $34.95.

Broderbund Software Inc.
1938 Fourth St.
San Rafael, CA 94901
(415) 456-6424

Super Mailer

This mailing program claims to have more features than any other mailing list available. Some of the standard features include creating and adding to the data file, sort by name, zip, or data line, and search on name or data line. Edit, modify, and printing any combination of entries are just a few more of the many standard features this program includes. Some of the Advance features include the merging of files, a wild card search, and the ability to use any printer. The program is written in Basic with machine language subroutines, and is written to be very easy to use. The program requires 48K, Basic, and 1 Disk Drive (2 Drives optional). The suggested retail price of this program is $49.95.

Royal Software
2160 West 11th Avenue
Eugene, Oregon 97402
(503) 683-5361

Thinking Game From Avalon Hill

Avalon Hill has released a computer version of its popular Facts In Five game. Computer Facts is a game of knowledge which puts players against time and each other. It includes options for solitaire play, doubles play, and party play.

In each round of the game five subject categories are selected from a list of more than 1000. Five letters are associated with each category, and the players supply answers that start with the designated letters.

Game difficulty can be controlled, modes for family and education are available. The sand clock timer and scoring system add to the challenge. This new thinking game from Avalon Hill requires 46K and its retail price is $26.

Avalon Hill Game Company
4517 Hartford Road
Baltimore, MD 21214

New Products cont'd.

Air Combat Game

Wingman is a combat simulator that allows one to four players to fly their fighter jets to defend their own territory while attacking and destroying the territory of the enemy. The game uses split screen techniques to display the battle positions of the two flight formation leaders. Each aircraft can be loaded with bombs, guns, or missiles, depending on the particular combat strategic mission. The winning combat pilots are the flight formation team that has the most net points from both offensive and defensive action.

The game territory consists of a European ground scenario, constructed randomly with each new game, that is 11 screens wide and contains airfields, fuel dumps, radars, military concentrations, and anti-aircraft guns and missiles.

Wingman is a action arcade type game that requires strategic combat planning to obtain an Allied victory. This game from MicroProse requires 40K Disk or Cassete and sells for $34.95.

Microprose Software
One Caribou Court
Parkton, Maryland 21120

Raving Reviewer cont'd.

the inevitable pile of manure lurking. Corner kicks and throw-ins almost invariably favor the defending team. The goalie is practically incapable of saving a shot from inside of the eighteen yard box and other(admittedly nit-picking) problems.

Fortunely these problems are shared by both teams, so they do not spoil the game.

SOCCER
Playability:8
Challenge:9
Graphics:9.5
Sound:6
Documentation:8
Overall Rating:8.5

Jake T.S.D. cont'd.

is good, there are a few complaints I have in this area. Firstly, the explosion of the Japanese aircraft is very questionable. It seems as if the whole program stops to see the aircraft explode. Secondly, the exact simularities between all the Japanese aircraft is a bit dissapointing. The programmer should have made the scout planes, bombers and zeroes three different shapes. Besides these few complaints I found the fast moving graphics to be great.

The sound of Hellcat Ace was quite good. The main attraction here is how you can hear the sound of your engine rev higher as you change speed.

The documentation of the game was great! I especially enjoyed the scenario write ups on the battles you encounter.

Overall I would definitely recommend Hellcat Ace to anyone. Its the type of game you'll have many "all nighters"(No one night stand stuff!). Furthermore, the way I figure it, it is better for you to fight W.W.II on the screen rather than watching some dumb actor do it for you!

HELLCAT ACE
MicroProse Software
One Caribou Court
Parkton, Maryland 21120
Playability:8
Challenge:9
Graphics:7.5
Sound:7.5
Documentation:8.5
Overall:8.2

Interview cont'd.

Q. What new programs does Adventure International have planned for the near future?

A. We have all sorts of new programs under development.

Q. What direction do you feel computer games are going?

A. Well we have got more sophisticated, and the days of '3D tic tac toe' and 'name that song' are long gone.

Titan Lander

by Bob Cockroft

Need 48K

Cruising 887,550,000 miles from the sun in your Phobus class landing vehicle, you have just entered the outer extremities of the Saturnian system. Ahead lies the giant ring planet with its multiple bands of toxic gases and its overwhelming brilliance over the blackness of deep space behind. To the right, you now see the disc of Saturn's outermost satellite, Phoebe. It's cold, barren surface sliding silently past your viewer and disappearing into the empty darkness which it emerged. Moving deeper and deeper into the Saturnian gravitational well, another object has just appeared on your viewer. It is the bright and protruding image of Lapetus. Like a giant iceball glowing in brilliance as it reflects the light rays from the distance Sun, lapetus's nature becomes apparent. It is a frozen, silent ball forever locked in a orbit around Saturn. Gliding away from this forbidden world, a warmer and more massive satellite is soon picked up on your viewer. After many months of space travel, the destination is in sight. The Saturn's largest Satellite, Titan is before you. As you get closer differences between Titan and other satellites become apparent. Unlike others, Titan has some atmosphere. Clouds of what looks like methane gas have obscured any surface features.

As your space craft automatically goes into an orbit around Titan, you review the instructions given to you on Earth by the project Supervisors. The instruction say to land at ten predetermined positions on the surface of Titan thus allowing the ships' computer to automatically perform a number of tests. The locations of the landing sights will be displayed by the computer as reddish rectangles.

All of what you do on Titan will be carefully analyzed by thousands of people who have worked for this landing project. Therefore, the performance you do could determine your future position in the space program. The project Supervisors must have confidence in your space piloting abilities if you are to be promoted or continue at your present assignment.

The most important thing to do in order to gain confidence would be to successfully land the ship. After reading the landing instructions, you know that the landing sights the project Supervisors have chosen, get progressively harder. More confidence will be gained for landing a harder sight than a easier one. Their are ten different levels and if you make it to the tenth the'll be a special surprise.

It cost millions of dollars to build one of the Phobus landing vehicles you are piloting. Therefore project Supervisors are looking for a pilot who can land the vehicle with minimum wear and tear. In other words, you must land both with little downward movement and with little horizontal movement.

The project Supervisors are always looking for perfection. A pin point landing is held with much esteem. Much confidence would be gained for this type of preformance. You have been told that the landing sights are larger than what the vehicle needs. A landing in the centre of one of these would be beneficial.

After landing the vehicle press the START BUTTON to see how the project Supervisors rated your landing. Also when starting a level it is important that you apply an upward thrust or you may run into an asteroid before you even get started.

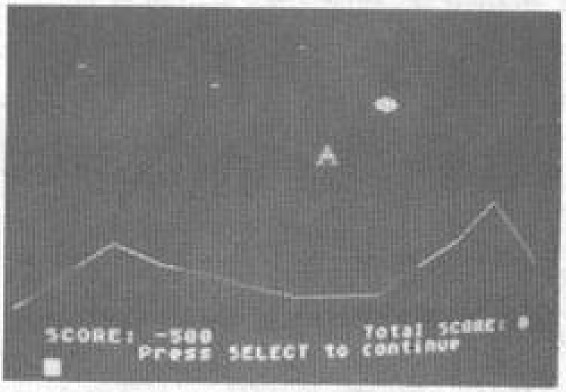
If you don't have time to type in the program below then send $10.00 for the disk and $6.00 for cassete version. Or send a blank disk or cassete and $5.00 to:

R.O.M.
P.O. BOX 252
MAPLE RIDGE, B.C. V2X 7G1

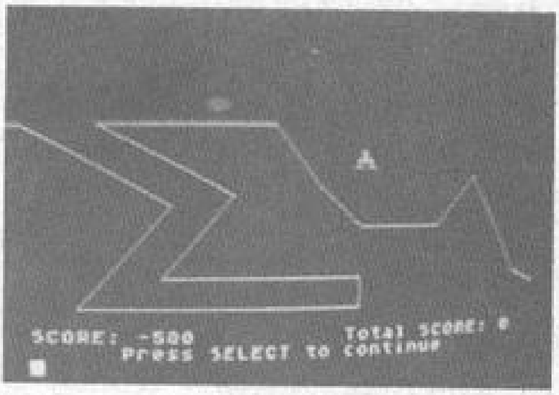
Remember this version of 'Titan Lander' is a little different then the one listed in this magazine. It has two more levels and more game features.

```
 1 REM ***** Titan Lander *****
 2 REM ***** by Bob Cockroft *****
 3 REM ***** ROM Magazine *****
 4 DIM YN$(5):GOTO 16000
 9 SC=0:TSC1=0:TSC2=0:TSC3=0:TSC4=0
:TSC=0:LIFE=1
 10 GRAPHICS 1:COLOR 1:SETCOLOR 2,16,1
 12 POKE 1589,40:POKE 1590,4:POKE
1591,13:POKE 1574,0
```

Titan Lander cont'd.



SCORE: -500 Total SCORE: 0
Press SELECT to continue



SCORE: -500 Total SCORE: 0
Press SELECT to continue

```
 16 FOR RR=1 TO 75:SOUND 1,162,10,10:SOUND 0,81,10,10:NEXT RR
 17 POSITION 4,5:? #6;"titan lander":FOR RR=1 TO 25:SOUND 0,144,10,10:SOUND 1,72,10,10:NEXT RR
 18 POSITION 9,10:? #6;"by":FOR RR=1 TO 50:SOUND 0,136,10,10:SOUND 1,68,10,10:NEXT RR
 19 FOR RR=1 TO 25:SOUND 0,162,10,10:SOUND 1,81,10,10:NEXT RR
 20 POSITION 4,12:? #6;"BOB COCKROFT":FOR RR=1 TO 75:SOUND 0,114,10,15:SOUND 1,57,10,15:POKE 708,RR:NEXT RR
 21 SOUND 0,1,1,0:SOUND 1,1,1,0:FOR RR=1 TO 700:NEXT RR
 29 I=PEEK(106)-40:J3=I*256+1792
 30 S=1
 32 FOR WW=1 TO 10:NEXT WW
 33 PL=1:GRAPHICS 1:SETCOLOR 2,16,1:POSITION 4,17:? #6;"PRESS start":POSITION 4,19:? #6;"TO CONTINUE"
 34 POSITION 1,5:? #6;"HOW MANY PLAYERS?":POSITION 1,7:? #6;"USE select BUTTON"
 35 IF PEEK(53279)=5 THEN PL=PL+1
 36 IF PL=5 THEN PL=1
 37 POSITION 18,5:? #6;PL
 38 FOR PLL=1 TO 50:NEXT PLL:IF PEEK(53279)<>6 THEN 35
 39 POSITION 2,10:? #6;"HOW MANY SHIPS?"
 40 IF PEEK(53279)=5 THEN LIFE=LIFE+1
 42 IF LIFE=6 THEN LIFE=1
 44 POSITION 17,10:? #6;LIFE
 46 FOR PLL=1 TO 50:NEXT PLL:IF PEEK(53279)<>6 THEN 39
 48 LIFE1=LIFE:LIFE2=LIFE:LIFE3=LIFE:LIFE4=LIFE
 70 FOR USE=1 TO PL
 72 IF USE=1 AND LIFE1<>0 THEN 80
 73 IF USE=2 AND LIFE2<>0 THEN 80
 74 IF USE=3 AND LIFE3<>0 THEN 80
 75 IF USE=4 AND LIFE4<>0 THEN 80
 77 GOTO 8990
 80 GRAPHICS 1:SETCOLOR 2,16,1:POSITION 3,9:? #6;"READY PLAYER ";USE:POSITION 6,12:? #6;"LEVEL ";S
 82 FOR HL=1 TO 75
 83 IF USE=1 THEN POSITION 6,14:? #6;"SHIPS: ";LIFE1
 84 IF USE=2 THEN POSITION 6,14:? #6;"SHIPS: ";LIFE2
 85 IF USE=3 THEN POSITION 6,14:? #6;"SHIPS: ";LIFE3
 86 IF USE=4 THEN POSITION 6,14:? #6;"SHIPS: ";LIFE4
 89 NEXT HL
 90 IF S=1 THEN 100
 91 IF S=2 THEN 200
 92 IF S=3 THEN 300
 93 IF S=10 THEN 1000
             94 IF S=4 THEN 400
 95 IF S=5 THEN 500
 96 IF S=6 THEN 600
 97 IF S=7 THEN 700
```

Basic Listing

```
98 IF S=8 THEN 800
99 IF S=9 THEN 900
100 REM ** SCREEN 1 **
102 HORZ3=130
105 VERT3=175
110 SIZE3=255
115 MAXUP=50
120 MAXDOWN=250
125 STREN=15
130 MAXAST=150
135 MAXAST1=88
199 GOTO 5000
200 REM ** SCREEN 2 **
202 HORZ3=101
205 VERT3=185
210 SIZE3=255
215 MAXUP=52
220 MAXDOWN=200
225 STREN=14
230 MAXAST=125
235 MAXAST1=68
299 GOTO 5000
300 REM ** SCREEN 3 **
302 HORZ3=78
305 VERT3=160
310 SIZE3=255
315 MAXUP=60
320 MAXDOWN=198
325 STREN=14
330 MAXAST=120
335 MAXAST1=65
399 GOTO 5000
400 REM ** SCREEN 4 **
402 HORZ3=118
405 VERT3=180
410 SIZE3=255
415 MAXUP=64
420 MAXDOWN=197
425 STREN=14
430 MAXAST=110
435 MAXAST1=58
499 GOTO 5000
500 REM ** SCREEN 5 **
502 HORZ3=129
505 VERT3=166
510 SIZE3=255
515 MAXUP=68
520 MAXDOWN=190
525 STREN=12
530 MAXAST=95
535 MAXAST1=38
```

```
599 GOTO 5000
600 REM ** SCREEN 6 **
602 HORZ3=143
605 VERT3=143
610 SIZE3=255
615 MAXUP=75
620 MAXDOWN=180
625 STREN=12
630 MAXAST=80
635 MAXAST1=38
699 GOTO 5000
700 REM ** SCREEN 7 **
702 HORZ3=120
705 VERT3=182
710 SIZE3=255
715 MAXUP=75
720 MAXDOWN=179
725 STREN=14
730 MAXAST=70
735 MAXAST1=40
799 GOTO 5000
800 REM ** SCREEN 8 **
802 HORZ3=111
805 VERT3=187
810 SIZE3=255
815 MAXUP=75
820 MAXDOWN=179
825 STREN=14
830 MAXAST=65
835 MAXAST1=40
899 GOTO 5000
900 REM ** SCREEN 9 **
902 HORZ3=140
905 VERT3=180
910 SIZE3=255
915 MAXUP=75
920 MAXDOWN=175
925 STREN=12
930 MAXAST=60
935 MAXAST1=35
999 GOTO 5000
1000 REM ** SCREEN 10 **
1002 HORZ3=186
1005 VERT3=185
1010 SIZE3=255
1015 MAXUP=70
1020 MAXDOWN=170
1025 STREN=10
1030 MAXAST=10
1035 MAXAST1=30
1999 GOTO 5000
```

Basic Listing

```
5000 POKE 53248,0:POKE 53249,0:POKE 53250,0:POKE 53251,0
5005 POKE 53252,0:POKE 53253,0:POKE 53254,0:POKE 53255,0
5020 POKE 1681,HORZ3:POKE 1682,VERT3:POKE 1683,SIZE3:POKE 1584,MAXUP:POKE 1585,M
AXDOWN:POKE 1583,STREN
5025 POKE 1586,MAXAST:POKE 1587,MAXAST1
5500 IF S=1 THEN 6000
5505 IF S=2 THEN 6100
5510 IF S=3 THEN 6200
5520 IF S=4 THEN 6300
5530 IF S=5 THEN 6400
5540 IF S=6 THEN 6500
5550 IF S=7 THEN 6600
5560 IF S=8 THEN 6700
5570 IF S=9 THEN 6800
5580 IF S=10 THEN 6900
5999 REM **** SCREEN 1 ****
6000 GRAPHICS 8:SETCOLOR 2,16,1
6010 PLOT 1,150:DRAWTO 50,120:DRAWTO 75,130:DRAWTO 150,145:DRAWTO 200,145:DRAWTO
 250,120:DRAWTO 275,100
6020 DRAWTO 300,125:DRAWTO 319,150
6025 POSITION 1,20:? "  Press START BUTTON after landing"
6050 GOTO 8000
6099 REM **** SCREEN 2 ****
6100 GRAPHICS 8:SETCOLOR 2,16,1
6110 PLOT 1,130:DRAWTO 25,90:DRAWTO 50,130:DRAWTO 75,140:DRAWTO 100,155:DRAWTO 1
25,155:DRAWTO 150,125
6120 DRAWTO 155,125:DRAWTO 175,100:DRAWTO 250,100:DRAWTO 260,87:DRAWTO 265,82:DR
AWTO 280,95:DRAWTO 290,100
6130 DRAWTO 300,100:DRAWTO 319,80
6150 GOTO 8000
6199 REM **** SCREEN 3 ****
6200 GRAPHICS 8:SETCOLOR 2,16,1
6210 PLOT 1,120:DRAWTO 20,130:DRAWTO 40,150:DRAWTO 60,130:DRAWTO 77,130:DRAWTO 9
2,60:DRAWTO 150,110:DRAWTO 160,110
6220 DRAWTO 190,135:DRAWTO 225,140:DRAWTO 245,110:DRAWTO 260,100:DRAWTO 295,140:
DRAWTO 319,135
6250 GOTO 8000
6299 REM **** SCREEN 4 ****
6300 GRAPHICS 8:SETCOLOR 2,16,1
6310 PLOT 1,100:DRAWTO 30,130:DRAWTO 60,100:DRAWTO 115,75:DRAWTO 117,100:DRAWTO
140,150:DRAWTO 157,150:DRAWTO 190,78
6320 DRAWTO 210,90:DRAWTO 220,75:DRAWTO 250,125:DRAWTO 290,112:DRAWTO 300,125:DR
AWTO 319,145
6350 GOTO 8000
6399 REM **** SCREEN 5 ****
6400 GRAPHICS 8:SETCOLOR 2,16,1
6410 PLOT 1,130:DRAWTO 10,120:DRAWTO 60,100:DRAWTO 100,130:DRAWTO 150,120:DRAWTO
 160,145:DRAWTO 180,145
6420 DRAWTO 200,120:DRAWTO 200,115:DRAWTO 190,100:DRAWTO 140,90:DRAWTO 145,80:DR
AWTO 200,75:DRAWTO 275,100
6425 DRAWTO 319,90
6430 POKE J3+VERT3+1,129:POKE J3+VERT3+2,195:POKE J3+VERT3+3,165:POKE J3+VERT3+4
,153:POKE J3+VERT3+5,153
6431 POKE J3+VERT3+6,165:POKE J3+VERT3+7,195:POKE J3+VERT3+8,129:POKE J3+VERT3+9
,129:POKE J3+VERT3+10,129
6450 GOTO 8000
```

Basic Listing

```
6499 REM **** SCREEN 6 ****
6500 GRAPHICS 8:SETCOLOR 2,16,1
6510 PLOT 1,100:DRAWTO 50,120:DRAWTO 75,110:DRAWTO 100,110:DRAWTO 115,140:DRAWTO
 130,150
6520 DRAWTO 210,150:DRAWTO 215,100:DRAWTO 210,90:DRAWTO 175,90:DRAWTO 160,75:DRA
WTO 160,130
6530 DRAWTO 140,130:DRAWTO 120,100:DRAWTO 125,80:DRAWTO 140,60:DRAWTO 250,60:DRA
WTO 300,75:DRAWTO 319,90
6540 PLOT 190,150:DRAWTO 190,110:PLOT 207,150:DRAWTO 207,110:PLOT 190,140:DRAWTO
 207,140
6541 PLOT 190,130:DRAWTO 207,130:PLOT 190,120:DRAWTO 207,120
6550 GOTO 8000
6599 REM **** SCREEN 7 ****
6600 GRAPHICS 8:SETCOLOR 2,16,1
6608 PLOT 1,100:DRAWTO 50,130
6609 DRAWTO 100,90:DRAWTO 180,75:DRAWTO 185,80:DRAWTO 170,86:DRAWTO 170,95:DRAWT
O 170,105:DRAWTO 165,110
6610 DRAWTO 143,152:DRAWTO 162,152:DRAWTO 182,125
6620 DRAWTO 213,80:DRAWTO 222,100:DRAWTO 250,130:DRAWTO 300,140
6650 GOTO 8000
6699 REM **** SCREEN 8 ****
6700 GRAPHICS 8:SETCOLOR 2,16,1
6710 PLOT 1,80:DRAWTO 50,120:DRAWTO 75,130:DRAWTO 80,110:DRAWTO 100,100:DRAWTO 1
40,75:DRAWTO 170,60:DRAWTO 200,80
6720 DRAWTO 200,100:DRAWTO 210,125:DRAWTO 175,120:DRAWTO 150,100:DRAWTO 125,100:
DRAWTO 125,155
6730 DRAWTO 145,155:DRAWTO 145,125:DRAWTO 175,145:DRAWTO 230,145:DRAWTO 236,100:
DRAWTO 300,90:DRAWTO 317,60
6750 GOTO 8000
6799 REM **** SCREEN 9 ****
6800 GRAPHICS 8:SETCOLOR 2,16,1
6810 PLOT 1,60:DRAWTO 2,60:DRAWTO 90,100:DRAWTO 40,150:DRAWTO 200,150:DRAWTO 200
,132:DRAWTO 88,132:DRAWTO 125,95
6812 DRAWTO 55,60
6820 DRAWTO 150,60:DRAWTO 175,90:DRAWTO 200,110:DRAWTO 250,110:DRAWTO 275,85:DRA
WTO 305,135:DRAWTO 319,140
6850 GOTO 8000
6899 REM **** SCREEN 10 ****
6900 GRAPHICS 8:SETCOLOR 2,16,1
6910 PLOT 1,60:DRAWTO 150,60:DRAWTO 20,70:DRAWTO 20,150:DRAWTO 250,150:DRAWTO 25
0,90:DRAWTO 275,90
6912 DRAWTO 275,155:DRAWTO 293,155
6920 DRAWTO 293,75:DRAWTO 232,75:DRAWTO 232,135:DRAWTO 38,135:DRAWTO 38,83:DRAWT
O 319,60
6950 GOTO 8000
7999 POKE 53278,1
8000 POKE 1536+8,0
8005 X=USR(26000)
8020 FOR X=VERT3 TO VERT3+50:POKE J3+X,0:NEXT X
8040 SCORE=PEEK(1576):SCORE1=PEEK(1577):SCORE2=PEEK(1578):SCORE3=PEEK(1580)
8050 SC=2*(SCORE+SCORE1+SCORE2+SCORE3)-1750+20*S
8051 IF SC<-250 THEN SC=-250
8055 IF PEEK(1574)=0 THEN L1=0
8056 IF PEEK(1574)=1 THEN L1=-1
8060 IF USE=1 THEN TSC1=TSC1+SC:TSC=TSC1:LIFE1=LIFE1+L1
8061 IF USE=2 THEN TSC2=TSC2+SC:TSC=TSC2:LIFE2=LIFE2+L1
8062 IF USE=3 THEN TSC3=TSC3+SC:TSC=TSC3:LIFE3=LIFE3+L1
```

Basic Listing

```
8063 IF USE=4 THEN TSC4=TSC4+SC:TSC=TSC4:LIFE4=LIFE4+L1
8064 ? "SCORE: ";:? SC;:? "                 Total SCORE: ";:? TSC
8075 ? "    Press SPACE BAR to continue"
8080 IF PEEK(764)<>33 THEN 8080
8081 POKE 764,65
8085 POKE 53248,0:POKE 53249,0:POKE 53250,0:POKE 53251,0:POKE 53252,0:POKE 53253
,0:POKE 53254,0:POKE 53255,0
8086 IF L1=0 AND S>9 THEN GOTO 15200
8087 IF L1=-1 THEN 72
8990 IF PL=1 AND LIFE1=0 THEN 15000
8991 IF PL=2 AND LIFE2=0 AND LIFE1=0 THEN 15000
8992 IF PL=3 AND LIFE1=0 AND LIFE2=0 AND LIFE3=0 THEN 15000
8993 IF PL=4 AND LIFE1=0 AND LIFE2=0 AND LIFE3=0 THEN 15000
8995 NEXT USE
8999 S=S+1
9000 GOTO 70
15000 GRAPHICS 0:SETCOLOR 2,16,1
15020 POSITION 16,3:? "RESULTS"
15030 POSITION 6,6:? ,"SCORE","  RANK"
15040 FOR PLAY=1 TO PL
15050 ? :? "  PLAYER ";PLAY;
15060 IF PLAY=1 THEN T=TSC1:? "     ";TSC1;
15065 IF PLAY=2 THEN T=TSC2:? "     ";TSC2;
15070 IF PLAY=3 THEN T=TSC3:? "     ";TSC3;
15075 IF PLAY=4 THEN T=TSC4:? "     ";TSC4;
15080 IF T<700 THEN ? "       Crater Maker":GOTO 15120
15085 IF T<800 THEN ? "       Cabin Boy   ":GOTO 15120
15090 IF T<900 THEN ? "       Rookie      ":GOTO 15120
15095 IF T<1000 THEN ? "       Scout      ":GOTO 15120
15100 IF T<1100 THEN ? "       Pilot      ":GOTO 15120
15105 IF T<1200 THEN ? "       Captain    ":GOTO 15120
15110 IF T<1300 THEN ? "       COMMANDER  ":GOTO 15120
15115 ? "      **  PRO  **     "
15120 ? :NEXT PLAY
15140 GOTO 15245
15200 GRAPHICS 1:POKE 710,50
15210 POSITION 2,4:? #6;"CONGRADULATIONS !"
15220 POSITION 4,8:? #6;"You     have        SUCCESSFULLY     completed all the
 levels.  ";
15230 ? #6;"You are nowthe Titan lander PRO"
15240 POSITION 8,15:? #6;"YeH!"
15245 REM
15247 ? :? "Would you like to play again?"
15248 INPUT YN$
15250 IF YN$="Y" THEN 9
15255 GOTO 15250
16000 GRAPHICS 0:SETCOLOR 2,16,1:? :? "The machine code will take a few       min
utes to load in."
16010 FOR HH=1 TO 1933:READ W:POKE 26000+HH-1,W:NEXT HH
16020 GOTO 9
16100 DATA 104,169,1,141,30,208,169,0,141,38,6,169,62,141,47,2,169,182,141,192,2
,165,106,56
16102 DATA 233,40,141,7,212,169,3,141,29,208,173,145,6,141,3,208,173,147,6,172,1
46,6,153,0
16104 DATA 127,153,1,127,169,255,141,20,6,141,21,6,169,0,141,9,6,141,7,6,141,17,
6,141
16106 DATA 18,6,141,5,6,141,0,6,141,4,6,141,16,6,141,23,6,141,36,6,141,37,6,141
16108 DATA 38,6,141,39,6,141,22,6,141,8,6,141,40,6,141,41,6,141,42,6,141,43,6,14
1
```

Basic Listing

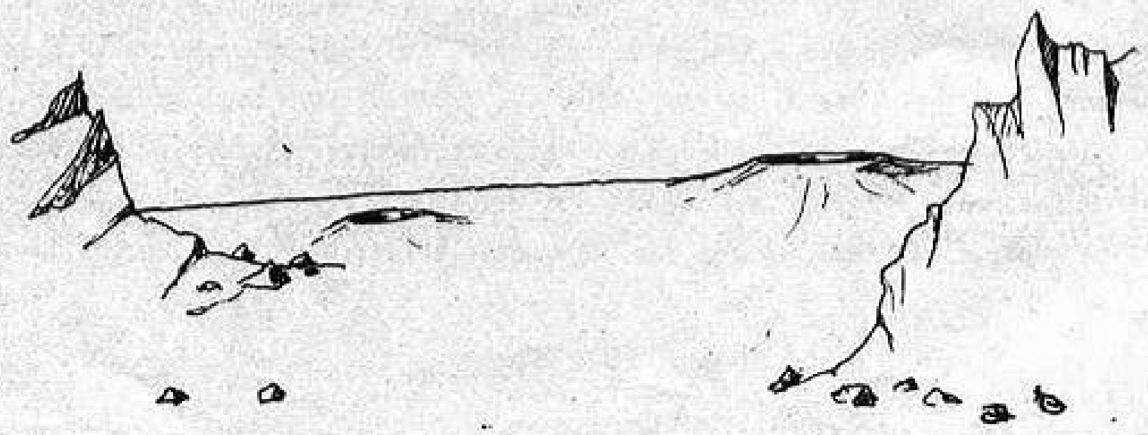
```
16110 DATA 44,6,141,45,6,141,46,6,169,5,141,6,6,141,19,6,169,20,141,3,6,169,245,
141
16112 DATA 2,6,173,10,210,141,32,6,169,1,141,1,6,141,52,6,169,191,141,193,2,173,
10,210
16114 DATA 141,34,6,173,10,210,141,33,6,169,26,141,194,2,169,55,141,195,2,173,10
,210,141,35
16116 DATA 6,173,52,6,201,0,240,17,238,52,6,201,255,208,10,169,0,141,52,6,169,14
0,141,2
16118 DATA 6,169,3,141,55,123,169,48,141,65,123,169,192,141,75,123,169,24,160,79
,153,1,126,153
16120 DATA 7,126,169,126,153,2,126,153,3,126,153,5,126,153,6,126,169,255,153,4,1
26,238,24,6
16122 DATA 173,24,6,205,51,6,208,41,169,0,141,24,6,173,32,6,201,215,208,5,169,10
,141,32
16124 DATA 6,238,32,6,141,2,208,173,33,6,201,40,208,5,169,210,141,33,6,206,33,6,
141,4
16126 DATA 208,238,25,6,173,25,6,205,51,6,208,41,169,0,141,25,6,173,34,6,201,215
,208,5
16128 DATA 169,30,141,34,6,238,34,6,141,6,208,173,35,6,201,40,208,5,169,215,141,
35,6,206
16130 DATA 35,6,141,7,208,173,38,6,201,1,208,11,162,0,232,236,53,6,208,250,76,25
1,106,173
16132 DATA 43,6,201,0,240,11,162,0,232,236,53,6,208,250,76,147,108,172,3,6,173,2
,6,141
16134 DATA 0,208,169,0,153,3,123,153,5,123,153,6,125,169,0,153,254,123,153,9,124
,169,8,153
16136 DATA 255,123,153,0,124,153,2,124,169,28,153,1,124,153,3,124,169,62,153,4,1
24,153,5,124
16138 DATA 169,42,153,6,124,169,34,153,7,124,169,117,153,8,124,173,0,6,201,0,208
,36,162,0
16140 DATA 232,236,54,6,208,250,173,1,6,201,0,209,11,162,0,232,236,55,6,208,250,
76,126,104
16142 DATA 201,2,208,3,76,53,104,76,91,104,172,3,6,173,2,6,141,1,208,173,4,6,201
,1
16144 DATA 240,49,201,2,240,61,201,0,240,3,76,254,103,169,63,141,193,2,169,62,15
3,7,125,153
16146 DATA 8,125,169,24,153,9,125,153,10,125,169,16,153,11,125,153,12,125,153,13
,125,238,4,6
16148 DATA 76,118,103,169,59,141,193,2,169,8,153,10,125,238,4,6,76,118,103,169,6
1,141,193,2
16150 DATA 169,24,153,10,125,169,4,153,11,125,153,12,125,153,13,125,238,4,6,76,1
18,103,169,0
16152 DATA 153,7,125,153,8,125,153,9,125,153,10,125,153,11,125,153,12,125,153,13
,125,153,14,125
16154 DATA 153,15,125,153,16,125,153,17,125,153,4,125,153,5,125,153,6,125,169,0,
141,4,6,141
16156 DATA 0,6,76,118,103,173,2,6,24,105,7,141,5,208,172,3,6,174,22,6,224,0,240,
8
16158 DATA 206,22,6,169,0,76,85,104,238,22,6,169,12,153,4,123,76,126,104,174,2,6
,142,5
16160 DATA 208,172,3,6,174,22,6,224,0,240,8,206,22,6,169,12,76,120,104,238,22,6,
169,0
16162 DATA 153,4,123,76,126,104,169,0,141,1,6,173,120,2,201,14,208,18,169,143,14
1,1,210,169
```

Basic Listing

```
16208 DATA 18,6,240,14,76,189,106,169,255,141,21,6,206,2,6,76,189,106,169,255,14
1,21,6,238
16210 DATA 2,6,76,189,106,173,38,6,201,0,240,3,76,251,106,173,4,208,201,0,208,45
,173,8
16212 DATA 208,201,0,208,38,173,10,208,201,0,208,31,173,11,208,201,0,208,24,173,
14,208,201,0
16214 DATA 208,17,173,12,208,201,0,240,7,201,10,240,3,76,81,102,76,147,108,169,0
,141,193,2
16216 DATA 169,1,141,38,6,173,37,6,201,10,208,1,96,238,39,6,174,39,6,224,5,240,3
,76
16218 DATA 81,102,162,0,142,39,6,201,0,240,3,76,51,108,238,36,6,173,36,6,201,1,2
08,3
16164 DATA 75,141,0,210,169,1,141,0,6,76,235,104,201,10,208,8,169,2,141,1,6,76,1
38,104
16166 DATA 201,6,208,8,169,1,141,1,6,76,138,104,201,11,208,18,169,2,141,1,6,169,
50,141
16168 DATA 0,210,169,143,141,1,210,76,70,105,201,7,208,18,169,1,141,1,6,169,50,1
41,0,210
16170 DATA 169,143,141,1,210,76,70,105,169,0,141,0,210,141,1,210,76,70,105,173,9
,6,201,0
16172 DATA 240,39,169,1,141,9,6,238,7,6,173,7,6,236,48,6,208,20,162,0,142,7,6,16
9
16174 DATA 1,76,7,105,173,5,6,201,250,240,3,238,5,6,76,153,105,169,0,141,9,6,238
,8
16176 DATA 6,174,8,6,236,48,6,208,15,162,0,142,8,6,173,6,6,201,0,240,6,206,6,6
16178 DATA 76,153,105,169,1,141,9,6,141,5,6,76,153,105,173,9,6,201,0,240,42,169,
1,141
16180 DATA 9,6,238,7,6,174,7,6,236,49,6,208,15,162,0,142,7,6,173,5,6,201,0,240
16182 DATA 6,206,5,6,76,153,105,169,0,141,9,6,76,153,105,169,0,141,9,6,238,8,6,1
74
16184 DATA 8,6,236,49,6,208,15,162,0,142,8,6,173,6,6,201,253,240,3,238,6,6,76,15
3
16186 DATA 105,173,1,6,201,0,208,3,76,79,106,201,1,208,3,76,254,105,76,173,105,1
73,16,6
16188 DATA 201,0,240,41,169,1,141,16,6,238,17,6,174,17,6,224,50,208,15,162,0,142
,17,6
16190 DATA 173,18,6,201,0,240,6,206,18,6,76,79,106,169,0,141,16,6,76,79,106,169,
0,141
16192 DATA 16,6,238,17,6,174,17,6,224,50,208,15,162,0,142,17,6,173,19,6,201,150,
240,3
16194 DATA 238,19,6,76,79,106,173,16,6,201,0,240,33,169,1,141,16,6,238,17,6,174,
17,6
16196 DATA 224,50,208,15,162,0,142,17,6,173,18,6,201,150,240,3,238,18,6,76,79,10
6,169,0
16198 DATA 141,16,6,238,17,6,174,17,6,224,50,208,15,162,0,142,17,6,173,19,6,201,
0,240
16200 DATA 6,206,19,6,76,79,106,169,1,141,16,6,76,79,106,173,38,6,201,1,208,3,76
,251
16202 DATA 106,206,20,6,173,20,6,205,5,6,240,8,205,6,6,240,14,76,130,106,169,255
,141,20
16204 DATA 6,206,3,6,76,130,106,169,255,141,20,6,238,3,6,76,130,106,238,23,6,173
,23,6
16206 DATA 201,5,240,3,76,189,106,169,0,141,23,6,206,21,6,173,21,6,205,19,6,240,
8,205
```

Basic Listing

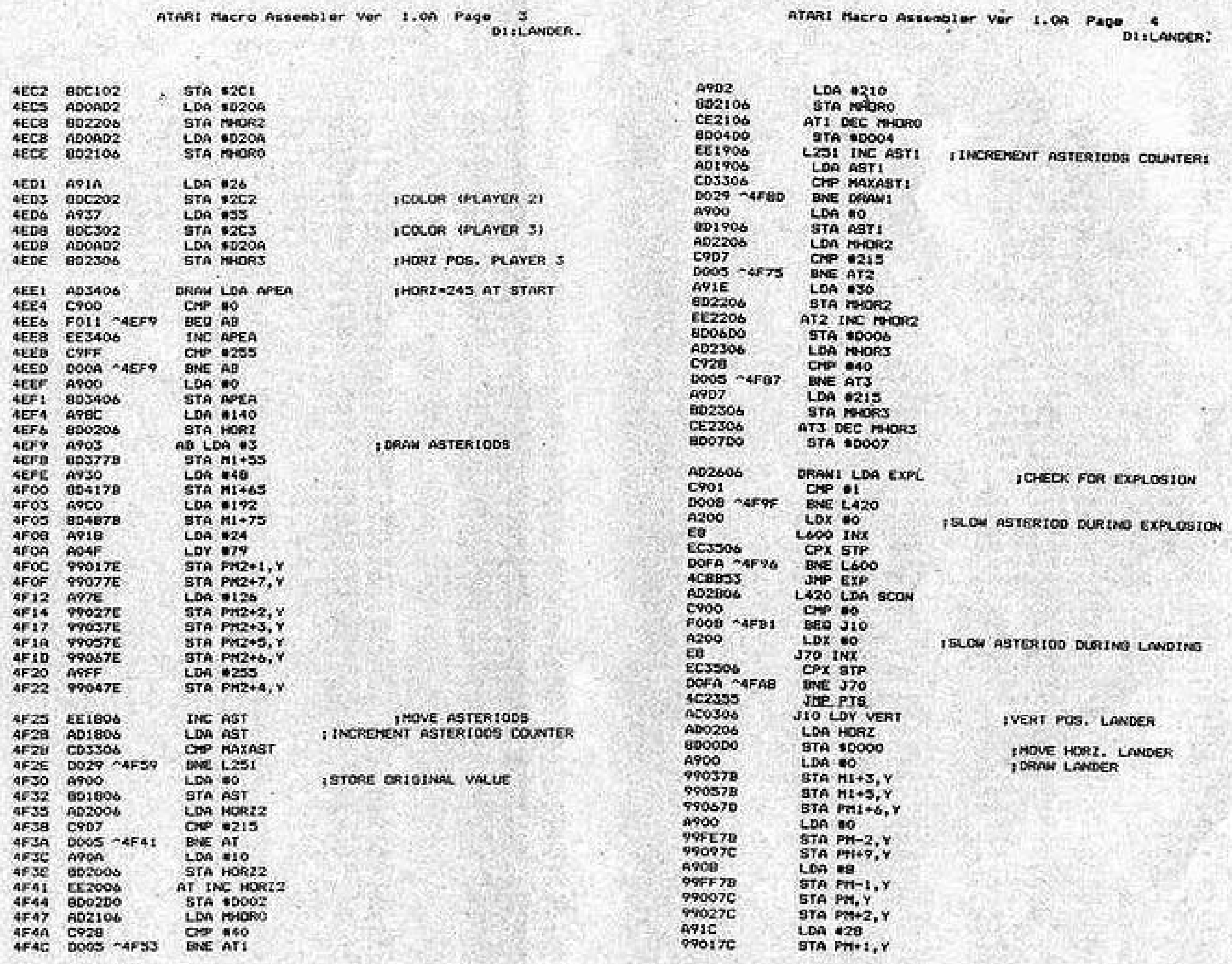
```
16220 DATA 76.82,107,201,75,208,3,76,147,107,201,150,208,3,76,201,107,201,175,20
8,3,76,253,107
16222 DATA 201,250,208,3,76,51,108,76,81,102,169,222,141,192,2,172,3,6,169,0,153
,254,123,153
16224 DATA 9,124,153,255,123,153,0,124,153,8,124,153,7,124,153,1,124,169,42,153,
2,124,153.6
16226 DATA 124,169,28,153,3,124,153,5,124,169,126,153,4,124,169,24,141,0,210,169
,14,141.1,210
16228 DATA 76,81,102,169,11,141,192,2,172,3,6,169,42,153,1,124,169,73,153,2,124,
169,28,153
16230 DATA 3,124,153,5,124,169,247,153,4,124,169,8,153,6,124,169,34,153,7,124,16
9,32,141,0
16232 DATA 210,169,14,141,1,210,76,81,102,169,53,141,192,2,172,3,6,169,24,153,1,
124,153.6
16234 DATA 124,169,36,153,2,124,153,5,124,169,67,153,3,124,169,194,153,4,124,169
,8,153.7,124
16236 DATA 169,100,141,0,210,169,5,141,1,210,76,81,102,169,56,141,192,2,172,3,6,
169,66,153
16238 DATA 0,124,153,7,124,169,129,153,1,124,153,6,124,169,0,153,2,124,153,3,124
,153,4,124
16240 DATA 153,5,124,153,8,124,169,60,141,0,210,169,10,141,1,210,76,81,102,172,3
,6,169,0
16242 DATA 153,254,123,153,255,123,153,0,124,153,1,124,153,2,124,153,3,124,153,4
,124,153.5,124
16244 DATA 153,6,124,153,7,124,153,8,124,153,3,123,153,4,123,153,5,123,153,5,125
,153,6,125
16246 DATA 153,7,125,153,8,125,153,9,125,153,10,125,153,11,125,153,12,125,153,13
,125,169,0,141
16248 DATA 0,210,141,1,210,238,37,6,173,43,6,201,0,240,1,96,76,81,102,173,6,6,56
,237
16250 DATA 47,6,48.3,76,251,106,173,43,6,201,0,240,13,173,31,208,201,6,208,3,76,
51,108
16252 DATA 76,81,102,169,1,141,43,6,169,255,56,237,6,6,141,40,6,169,255,56,237,1
8,6,141
16254 DATA 41,6,169,255,56,237,19,6,141,44,6,173,2,6,56,237,145,6,201,0,208,8,16
9,255
16256 DATA 141,46,6,76.241,108,141,45,6,48,6,169,255,56,237,45,6,141,42,6,169,0,
172,3
16258 DATA 6,153,7,125,153,8,125,153,9,125,153,10,125,153,11,125,153,12,125,153,
13,125,141,0
16260 DATA 210,141,1,210,153,3,123,153,4,123,76,81,102
```



```
                         ATARI Macro Assembler Ver  1.0A  Page   1
                                                        D1:LANDER.

4E20  = 4E20       ORG $6590
4E20  68           PLA
4E21  A901         LDA #1                  ;CLEAR COLLISIONS
4E23  8D1ED0       STA $D01E
4E26  A900         LDA #0
4E28  8D2606       STA $626
4E2B  A93E         LDA #62                 ;SET P/M GRAPHICS
4E2D  8D2F02       STA $22F
4E30  A9B6         LDA #182                ;COLOR (PLAYER 0)
4E32  8DC002       STA $2C0
4E35  A56A         LDA $6A
4E37  38           SEC
4E38  E928         SBC #40
4E3A  8D07D4       STA $D407
4E3D  A903         LDA #3
4E3F  8D1DD0       STA $D01D
      = 7C00      PM SET $7C00                 ;DECLARATION SECTION
      = 7B00      M1 SET $7B00           ;BASE LOCATION FOR MISSILES
      = 7D00      PM1 SET $7D00          ;BASE PLAYER 1
      = 7E00      PM2 SET $7E00          ;BASE PLAYER 2
      = 7F00      PM3 SET $7F00          ;BASE PLAYER 3
      = 0600      THRUST SET $600        ;THRUST FLAG
      = 0601      MICRO SET $601         ;SIDE THRUST FLAG
      = 0602      HORZ SET $602          ;HORZ POS LANDER
      = 0603      VERT SET $603          ;VERT POS LANDER
      = 0604      FAZE SET $604          ;THRUST TYPE
      = 0605      UMOVE SET $605         ;UP ACCELERATION
      = 0606      DMOVE SET $606         ;DOWN ACCELERATION
      = 0607      USPEED SET $607        ;UP COUNTER
      = 0608      DSPEED SET $608        ;DOWN COUNTER
      = 0609      UM SET $609            ;VERT DIRECTION FLAG
      = 0610      HUM SET $610           ;HORZ DIRECTION FLAG
      = 0611      UHSPD SET $611         ;HORZ ACCEL COUNTER
      = 0612      RMOVE SET $612         ;RIGHT ACCELERATION
      = 0613      LMOVE SET $613         ;LEFT ACCELERATION
      = 0614      COUNT SET $614         ;RATE COUNTER
      = 0615      COUNT1 SET $615        ;RATE COUNTER1
      = 0616      CHANGE SET $616        ;CHANGE SIDE THRUST
      = 0617      SLOW SET $617
      = 0618      AST SET $618           ;ASTERIOD COUNTER
      = 0619      AST1 SET $619          ;ASTERIOD COUNTER1
      = 0620      HORZ2 SET $620         ;HORZ POS PLAYER2
      = 0621      MHOR0 SET $621         ;HORZ POS MISSILE0
      = 0622      MHOR2 SET $622         ;HORZ POS MISSILE2
      = 0623      MHOR3 SET $623         ;HORZ POS MISSILE3
      = 0624      ECON SET $624          ;EXPLOSION TYPE
      = 0625      ECON1 SET $625         ;EXPLOSION LOOP
      = 0626      EXPL SET $626          ;EXPLOSION FLAG
      = 0627      EXCON SET $627         ;EXPLOSION SLOWER
      = 0628      SCORE SET $628         ;SCORE (VERT)
      = 0629      SCORE1 SET $629        ;SCORE (HORZ)R
      = 062A      SCORE2 SET $62A        ;SCORE (POS.)
      = 062B      SCON SET $62B          ;SCORE LOOP
      = 062C      SCORE3 SET $62C        ;SCORE (HORZ)L
      = D01F      CONSOL SET $D01F       ;CONSOL FLAG
      = 062D      STORE SET $62D         ;STORE VALUE
      = 062E      PIN SET $62E           ;PIN POINT LANDING FLAG
```

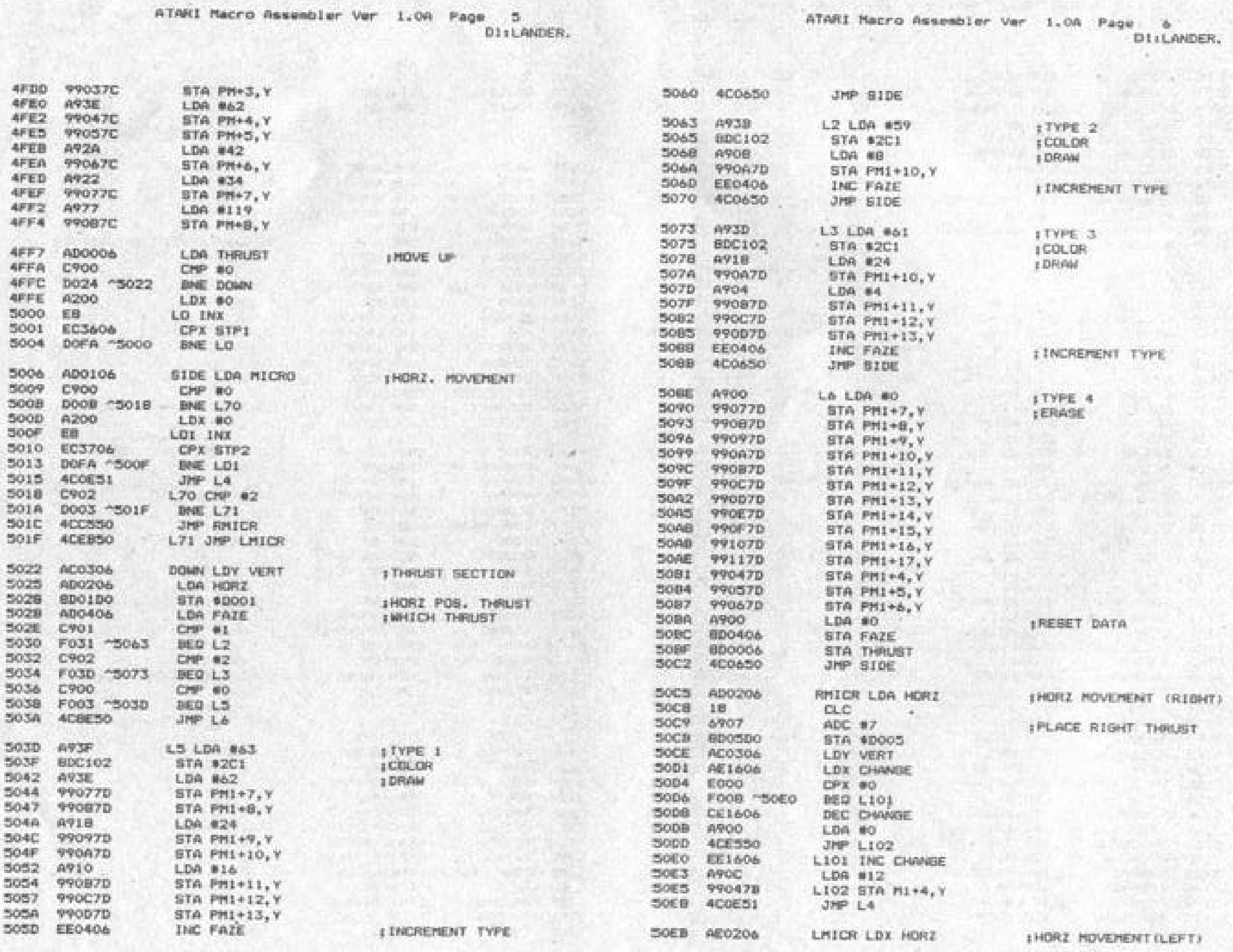
```
                         ATARI Macro Assembler Ver  1.0A  Page   2
                                                        D1:LANDER.

      = 062F       STREN SET $62F          ;STRENGTH OF LANDER
      = 0630       MAXUP SET $630          ;MAX ACCEL UP
      = 0631       MAXDOWN SET $631        ;MAX ACCEL DOWN
      = 0632       MAXAST SET $632         ;MAX SPEED ASTEROIDS(0)
      = 0633       MAXAST1 SET $633        ;MAX SPEED ASTEROIDS(1)
      = 0634       APEA SET $634           ;BEGIN WITH HORZ 245
      = 0635       STP SET $635            ;SLOW ASTERIODS
      = 0636       STP1 SET $636           ;SLOW ASTERIODS
      = 0637       STP2 SET $637           ;SLOW ASTERIODS
4E42  AD9106        LDA $691                    ;LANDING PAD
4E45  8D03D0        STA $D003                   ;HORZ POS.
4E48  AD9306        LDA $693                    ;SIZE
4E4B  AC9206        LDY $692                    ;VERT POS.
4E4E  99007F        STA PM3,Y
4E51  99017F        STA PM3+1,Y

                                           ;GIVE VALUES TO VARIABLES
4E54  A9FF          LDA #255
4E56  8D1406        STA COUNT
4E59  8D1506        STA COUNT1
4E5C  A900          LDA #0
4E5E  8D0906        STA UM
4E61  8D0706        STA USPEED
4E64  8D1106        STA UHSPD
4E67  8D1206        STA RMOVE
4E6A  8D0506        STA UMOVE
4E6D  8D0006        STA THRUST
4E70  8D0406        STA FAZE
4E73  8D1006        STA HUM
4E76  8D1706        STA SLOW
4E79  8D2406        STA ECON
4E7C  8D2506        STA ECON1
4E7F  8D2606        STA EXPL
4E82  8D2706        STA EXCON
4E85  8D1606        STA CHANGE
4E88  8D0806        STA DSPEED
4E8B  8D2806        STA SCORE
4E8E  8D2906        STA SCORE1
4E91  8D2A06        STA SCORE2
4E94  8D2B06        STA SCON
4E97  8D2C06        STA SCORE3
4E9A  8D2D06        STA STORE
4E9D  8D2E06        STA PIN
4EA0  A905          LDA #5
4EA2  8D0606        STA DMOVE
4EA5  8D1306        STA LMOVE
4EA8  A914          LDA #20
4EAA  8D0306        STA VERT
4EAD  A9F5          LDA #245
4EAF  8D0206        STA HORZ
4EB2  AD0AD2        LDA $D20A
4EB5  8D2006        STA HORZ2
4EB8  A901          LDA #1
4EBA  8D0106        STA MICRO
4EBD  8D3406        STA APEA
4EC0  A9BF          LDA #191                        ;COLOR (PLAYER1)
```

Assembly Listing



```
                         ATARI Macro Assembler Ver  1.0A  Page   3
                                                           D1:LANDER.

4EC2  8DC102         STA $2C1
4EC5  AD0AD2         LDA $D20A
4EC8  8D2206         STA MHOR2
4ECB  AD0AD2         LDA $D20A
4ECE  8D2106         STA MHOR0

4ED1  A91A           LDA #26
4ED3  8DC202         STA $2C2              ;COLOR (PLAYER 2)
4ED6  A937           LDA #55
4ED8  8DC302         STA $2C3              ;COLOR (PLAYER 3)
4EDB  AD0AD2         LDA $D20A
4EDE  8D2306         STA MHOR3             ;HORZ POS. PLAYER 3

4EE1  AD3406        DRAW LDA APEA          ;HORZ=245 AT START
4EE4  C900           CMP #0
4EE6  F011 ^4EF9     BEQ AB
4EE8  EE3406         INC APEA
4EEB  C9FF           CMP #255
4EED  D00A ^4EF9     BNE AB
4EEF  A900           LDA #0
4EF1  8D3406         STA APEA
4EF4  A98C           LDA #140
4EF6  8D0206         STA HORZ
4EF9  A903          AB LDA #3             ;DRAW ASTERIODS
4EFB  8D377B         STA M1+55
4EFE  A930           LDA #48
4F00  8D417B         STA M1+65
4F03  A9C0           LDA #192
4F05  8D4B7B         STA M1+75
4F08  A918           LDA #24
4F0A  A04F           LDY #79
4F0C  99017E         STA PM2+1,Y
4F0F  99077E         STA PM2+7,Y
4F12  A97E           LDA #126
4F14  99027E         STA PM2+2,Y
4F17  99037E         STA PM2+3,Y
4F1A  99057E         STA PM2+5,Y
4F1D  99067E         STA PM2+6,Y
4F20  A9FF           LDA #255
4F22  99047E         STA PM2+4,Y

4F25  EE1806         INC AST                   ;MOVE ASTERIODS
4F28  AD1806         LDA AST         ;INCREMENT ASTERIODS COUNTER
4F2B  CD3306         CMP MAXAST
4F2E  D029 ^4F59     BNE L251
4F30  A900           LDA #0          ;STORE ORIGINAL VALUE
4F32  8D1806         STA AST
4F35  AD2006         LDA HORZ2
4F38  C9D7           CMP #215
4F3A  D005 ^4F41     BNE AT
4F3C  A90A           LDA #10
4F3E  8D2006         STA HORZ2
4F41  EE2006        AT INC HORZ2
4F44  8D02D0         STA $D002
4F47  AD2106         LDA MHOR0
4F4A  C928           CMP #40
4F4C  D005 ^4F53     BNE AT1

                         ATARI Macro Assembler Ver  1.0A  Page   4
                                                           D1:LANDER:

      A9D2           LDA #210
      8D2106         STA MHOR0
      CE2106        AT1 DEC MHOR0
      8D04D0         STA $D004
      EE1906        L251 INC AST1     ;INCREMENT ASTERIODS COUNTER1
      AD1906         LDA AST1
      CD3306         CMP MAXAST1
      D029 ^4FBD     BNE DRAW1
      A900           LDA #0
      8D1906         STA AST1
      AD2206         LDA MHOR2
      C9D7           CMP #215
      D005 ^4F75     BNE AT2
      A91E           LDA #30
      8D2206         STA MHOR2
      EE2206        AT2 INC MHOR2
      8D06D0         STA $D006
      AD2306         LDA MHOR3
      C928           CMP #40
      D005 ^4F87     BNE AT3
      A9D7           LDA #215
      8D2306         STA MHOR3
      CE2306        AT3 DEC MHOR3
      8D07D0         STA $D007

      AD2606        DRAW1 LDA EXPL          ;CHECK FOR EXPLOSION
      C901           CMP #1
      D008 ^4F9F     BNE L420
      A200           LDX #0           ;SLOW ASTERIOD DURING EXPLOSION
      E8            L600 INX
      EC3506         CPX STP
      D0FA ^4F96     BNE L600
      4C8853         JMP EXP
      AD2806        L420 LDA SCON
      C900           CMP #0
      F008 ^4FB1     BEQ J10
      A200           LDX #0           ;SLOW ASTERIOD DURING LANDING
      E8            J70 INX
      EC3506         CPX STP
      D0FA ^4FAB     BNE J70
      4C2355         JMP PTS
      AC0306        J10 LDY VERT            ;VERT POS. LANDER
      AD0206         LDA HORZ
      8D00D0         STA $D000               ;MOVE HORZ. LANDER
      A900           LDA #0                  ;DRAW LANDER
      99037B         STA M1+3,Y
      99057B         STA M1+5,Y
      99067D         STA PM1+6,Y
      A900           LDA #0
      99FE7B         STA PM-2,Y
      99097C         STA PM+9,Y
      A908           LDA #8
      99FF7B         STA PM-1,Y
      99007C         STA PM,Y
      99027C         STA PM+2,Y
      A91C           LDA #28
      99017C         STA PM+1,Y
```

Assembly Listing



```
                ATARI Macro Assembler Ver  1.0A  Page   5
                                                  D1:LANDER.

4FDD  99037C           STA PM+3,Y
4FE0  A93E             LDA #62
4FE2  99047C           STA PM+4,Y
4FE5  99057C           STA PM+5,Y
4FE8  A92A             LDA #42
4FEA  99067C           STA PM+6,Y
4FED  A922             LDA #34
4FEF  99077C           STA PM+7,Y
4FF2  A977             LDA #119
4FF4  99087C           STA PM+8,Y

4FF7  AD0006           LDA THRUST                ;MOVE UP
4FFA  C900             CMP #0
4FFC  D024 ^5022       BNE DOWN
4FFE  A200             LDX #0
5000  E8              L0 INX
5001  EC3606           CPX STP1
5004  D0FA ^5000       BNE L0

5006  AD0106          SIDE LDA MICRO             ;HORZ. MOVEMENT
5009  C900             CMP #0
500B  D00B ^5018       BNE L70
500D  A200             LDX #0
500F  E8              L01 INX
5010  EC3706           CPX STP2
5013  D0FA ^500F       BNE L01
5015  4C0E51           JMP L4
5018  C902            L70 CMP #2
501A  D003 ^501F       BNE L71
501C  4CC550           JMP RMICR
501F  4CEB50          L71 JMP LMICR

5022  AC0306          DOWN LDY VERT              ;THRUST SECTION
5025  AD0206           LDA HORZ
5028  8D01D0           STA $D001                 ;HORZ POS. THRUST
502B  AD0406           LDA FAZE                  ;WHICH THRUST
502E  C901             CMP #1
5030  F031 ^5063       BEQ L2
5032  C902             CMP #2
5034  F03D ^5073       BEQ L3
5036  C900             CMP #0
5038  F003 ^503D       BEQ L5
503A  4C8E50           JMP L6

503D  A93F            L5 LDA #63                 ;TYPE 1
503F  8DC102           STA $2C1                  ;COLOR
5042  A93E             LDA #62                   ;DRAW
5044  99077D           STA PM1+7,Y
5047  99087D           STA PM1+8,Y
504A  A918             LDA #24
504C  99097D           STA PM1+9,Y
504F  990A7D           STA PM1+10,Y
5052  A910             LDA #16
5054  990B7D           STA PM1+11,Y
5057  990C7D           STA PM1+12,Y
505A  990D7D           STA PM1+13,Y
505D  EE0406           INC FAZE                  ;INCREMENT TYPE

                ATARI Macro Assembler Ver  1.0A  Page   6
                                                  D1:LANDER.

5060  4C0650           JMP SIDE

5063  A93B            L2 LDA #59                 ;TYPE 2
5065  8DC102           STA $2C1                  ;COLOR
5068  A908             LDA #8                    ;DRAW
506A  990A7D           STA PM1+10,Y
506D  EE0406           INC FAZE                  ;INCREMENT TYPE
5070  4C0650           JMP SIDE

5073  A93D            L3 LDA #61                 ;TYPE 3
5075  8DC102           STA $2C1                  ;COLOR
5078  A918             LDA #24                   ;DRAW
507A  990A7D           STA PM1+10,Y
507D  A904             LDA #4
507F  990B7D           STA PM1+11,Y
5082  990C7D           STA PM1+12,Y
5085  990D7D           STA PM1+13,Y
5088  EE0406           INC FAZE                  ;INCREMENT TYPE
508B  4C0650           JMP SIDE

508E  A900            L6 LDA #0                  ;TYPE 4
5090  99077D           STA PM1+7,Y               ;ERASE
5093  99087D           STA PM1+8,Y
5096  99097D           STA PM1+9,Y
5099  990A7D           STA PM1+10,Y
509C  990B7D           STA PM1+11,Y
509F  990C7D           STA PM1+12,Y
50A2  990D7D           STA PM1+13,Y
50A5  990E7D           STA PM1+14,Y
50A8  990F7D           STA PM1+15,Y
50AB  99107D           STA PM1+16,Y
50AE  99117D           STA PM1+17,Y
50B1  99047D           STA PM1+4,Y
50B4  99057D           STA PM1+5,Y
50B7  99067D           STA PM1+6,Y
50BA  A900             LDA #0                    ;RESET DATA
50BC  8D0406           STA FAZE
50BF  8D0006           STA THRUST
50C2  4C0650           JMP SIDE

50C5  AD0206          RMICR LDA HORZ             ;HORZ MOVEMENT (RIGHT)
50C8  18               CLC
50C9  6907             ADC #7                    ;PLACE RIGHT THRUST
50CB  8D05D0           STA $D005
50CE  AC0306           LDY VERT
50D1  AE1606           LDX CHANGE
50D4  E000             CPX #0
50D6  F008 ^50E0       BEQ L101
50D8  CE1606           DEC CHANGE
50DB  A900             LDA #0
50DD  4CE550           JMP L102
50E0  EE1606          L101 INC CHANGE
50E3  A90C             LDA #12
50E5  99047B          L102 STA M1+4,Y
50E8  4C0E51           JMP L4

50EB  AE0206          LMICR LDX HORZ             ;HORZ MOVEMENT(LEFT)
```

```
                    ATARI Macro Assembler Ver  1.0A  Page   7
                                                    D1:LANDER.

50EE  8E05D0        STX $D005            ;PLACE LEFT THRUST
50F1  AC0306        LDY VERT
50F4  AE1606        LDX CHANGE
50F7  E000          CPX #0
50F9  F008 ^5103    BEQ L103
50FB  CE1606        DEC CHANGE
50FE  A90C          LDA #12
5100  4C0851        JMP L104
5103  EE1606       L103 INC CHANGE
5106  A900          LDA #0
5108  990478       L104 STA M1+4,Y
510B  4C0E51        JMP L4

                              ;DETERMINE MOVEMENT VECTORS

510E  A900         L4 LDA #0
5110  8D0106        STA MICRO
5113  AD7802        LDA $278             ;LOAD STICK(0) POSITION
5116  C90E          CMP #14
5118  D012 ^512C    BNE L10
511A  A98F         L11 LDA #143          ;UP
511C  8D0102        STA $D201
511F  A94B          LDA #75
5121  8D0002        STA $D200
5124  A901          LDA #1
5126  8D0006        STA THRUST
5129  4C7B51        JMP EX
512C  C90A         L10 CMP #10           ;UP/RIGHT
512E  D008 ^5138    BNE L12
5130  A902          LDA #2
5132  8D0106        STA MICRO
5135  4C1A51        JMP L11
5138  C906         L12 CMP #6            ;UP/LEFT
513A  D008 ^5144    BNE L14
513C  A901          LDA #1
513E  8D0106        STA MICRO
5141  4C1A51        JMP L11
5144  C90B         L14 CMP #11           ;RIGHT
5146  D012 ^515A    BNE L15
5148  A902          LDA #2
514A  8D0106        STA MICRO
514D  A932          LDA #50
514F  8D0002        STA $D200
5152  A98F          LDA #143
5154  8D0102        STA $D201
5157  4CD651        JMP MA
515A  C907         L15 CMP #7            ;LEFT
515C  D012 ^5170    BNE L16
515E  A901          LDA #1
5160  8D0106        STA MICRO
5163  A932          LDA #50
5165  8D0002        STA $D200
5168  A98F          LDA #143
516A  8D0102        STA $D201
516D  4CD651        JMP MA
5170  A900         L16 LDA #0
5172  8D0002        STA $D200
```

```
                    ATARI Macro Assembler Ver  1.0A  Page   8
                                                    D1:LANDER.

5175  8D0102        STA $D201
5178  4CD651        JMP MA

                              ;CHANGE VECTOR (UP)

517B  AD0906       EX LDA UM              ;NOW MOVE UP
517E  C900          CMP #0
5180  F027 ^51A9    BEQ DMOV
5182  A901          LDA #1
5184  8D0906        STA UM
5187  EE0706        INC USPEED            ;ACCELERATION COUNTER
518A  AD0706        LDA USPEED
518D  EC3006        CPX MAXUP
5190  D014 ^51A6    BNE L25
5192  A200          LDX #0
5194  8E0706        STX USPEED
5197  A901         L100 LDA #1
5199  4C9751        JMP L100
519C  AD0506        LDA UMOVE
519F  C9FA          CMP #250
51A1  F003 ^51A6    BEQ L25
51A3  EE0506        INC UMOVE           ;INCREASE VERTICAL ACCEL.
51A6  4C2952       L25 JMP LR

51A9  A900         DMOV LDA #0            ;NOW MOVE DOWN
51AB  8D0906        STA UM
51AE  EE0806        INC DSPEED
51B1  AE0806        LDX DSPEED        ;ACCEL COUNTER
51B4  EC3006        CPX MAXUP
51B7  D00F-^51C8    BNE L27
51B9  A200          LDX #0
51BB  8E0806        STX DSPEED
51BE  AD0606        LDA DMOVE
51C1  C900          CMP #0
51C3  F006 ^51CB    BEQ L30
51C5  CE0606        DEC DMOVE
51C8  4C2952       L27 JMP LR
51CB  A901         L30 LDA #1
51CD  8D0906        STA UM
51D0  8D0506        STA UMOVE         ;DECREASE ACCEL. COUNTER
51D3  4C2952        JMP LR

                                  ;CHANGE VECTOR DOWN

51D6  AD0906       MA LDA UM            ;NOW MOVE UP
51D9  C900          CMP #0
51DB  F02A ^5207    BEQ DMOVD
51DD  A901          LDA #1
51DF  8D0906        STA UM
51E2  EE0706        INC USPEED        ;ACCEL. COUNTER
51E5  AE0706        LDX USPEED
51E8  EC3106        CPX MAXDOWN
51EB  D00F ^51FC    BNE L32
51ED  A200          LDX #0
51EF  8E0706        STX USPEED
51F2  AD0506        LDA UMOVE
51F5  C900          CMP #0
```

Assembly Listing

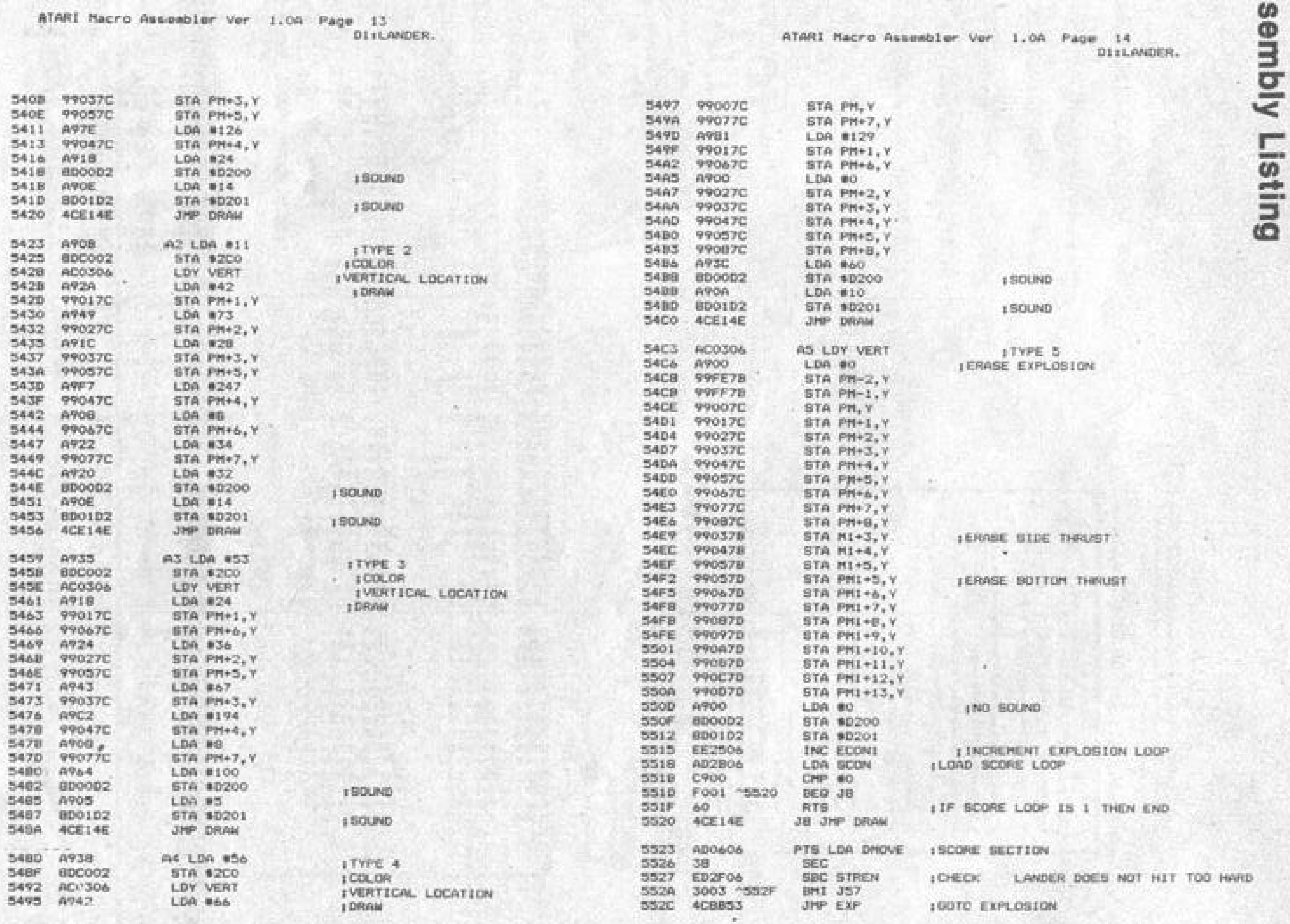
```
                ATARI Macro Assembler Ver  1.0A  Page   9
                                                D1:LANDER.

51F7  F006 ^51FF   BEQ L34
51F9  CE0506       DEC UMOVE        ;DECREASE ACCEL. COUNTER
51FC  4C2952      L32 JMP LR
51FF  A900        L34 LDA #0
5201  8D0906       STA UM
5204  4C2952       JMP LR

5207  A900        DMOVD LDA #0              ;NOW MOVE DOWN
5209  8D0906       STA UM
520C  EE0806       INC DSPEED       ;ACCEL. COUNTER
520F  AE0806       LDX DSPEED
5212  EC3106       CPX MAXDOWN
5215  D00F ^5226   BNE L35
5217  A200         LDX #0
5219  8E0806       STX DSPEED
521C  AD0606       LDA DMOVE
521F  C9FD         CMP #253
5221  F003 ^5226   BEQ L35
5223  EE0606       INC DMOVE        ;INCREASE ACCEL. COUNTER
5226  4C2952      L35 JMP LR

                                            ;HORIZONTAL MOVEMENT
5229  AD0106      LR LDA MICRO
522C  C900         CMP #0
522E  D003 ^5233   BNE L40
5230  4CDF52       JMP WAK
5233  C901        L40 CMP #1
5235  D003 ^523A   BNE L41
5237  4C8E52       JMP RHOR
523A  4C3D52      L41 JMP LHOR

                                            ;LEFT VECTOR
523D  AD1006      LHOR LDA HUM              ;RIGHT
5240  C900         CMP #0
5242  F029 ^526D   BEQ DHD
5244  A901         LDA #1
5246  8D1006       STA HUM
5249  EE1106       INC UHSPD        ;HORI ACCEL COUNTER
524C  AE1106       LDX UHSPD
524F  E032         CPX #50
5251  D00F ^5262   BNE L53
5253  A200         LDX #0
5255  8E1106       STX UHSPD
5258  AD1206       LDA RMOVE
525B  C900         CMP #0
525D  F006 ^5265   BEQ L54
525F  CE1206       DEC RMOVE        ;DECREASE HORI ACCEL
5262  4CDF52      L53 JMP WAK
5265  A900        L54 LDA #0
5267  8D1006       STA HUM
526A  4CDF52       JMP WAK

526D  A900        DHD LDA #0                ;LEFT
526F  8D1006       STA HUM
5272  EE1106       INC UHSPD        ;HORI ACCEL. COUNTER
5275  AE1106       LDX UHSPD
5278  E032         CPX #50
527A  D00F ^528B   BNE L57
```

```
                ATARI Macro Assembler Ver  1.0A  Page  10
                                                D1:LANDER.

527C  A200         LDX #0
527E  8E1106       STX UHSPD
5281  AD1306       LDA LMOVE
5284  C996         CMP #150
5286  F003 ^528B   BEQ L57
5288  EE1306       INC LMOVE     -  ;INCREASE HORI ACCEL.
528B  4CDF52      L57 JMP WAK

                                            ;MOVEMENT VECTOR RIGHT
528E  AD1006      RHOR LDA HUM              ;RIGHT
5291  C900         CMP #0
5293  F021 ^52B6   BEQ DHR
5295  A901         LDA #1
5297  8D1006       STA HUM
529A  EE1106       INC UHSPD         ;HORI ACCEL. COUNTER
529D  AE1106       LDX UHSPD
52A0  E032         CPX #50
52A2  D00F ^52B3   BNE L50
52A4  A200         LDX #0
52A6  8E1106       STX UHSPD
52A9  AD1206       LDA RMOVE
52AC  C996         CMP #150
52AE  F003 ^52B3   BEQ L50
52B0  EE1206       INC RMOVE         ;INCREASE HORI ACCEL.
52B3  4CDF52      L50 JMP WAK

52B6  A900        DHR LDA #0                ;LEFT
52B8  8D1006       STA HUM
52BB  EE1106       INC UHSPD        ;HORI ACCEL.
52BE  AE1106       LDX UHSPD
52C1  E032         CPX #50
52C3  D00F ^52D4   BNE L51
52C5  A200         LDX #0
52C7  8E1106       STX UHSPD
52CA  AD1306       LDA LMOVE
52CD  C900         CMP #0
52CF  F006 ^52D7   BEQ L52
52D1  CE1306       DEC LMOVE         ;DECRECE HORI ACCEL
52D4  4CDF52      L51 JMP WAK
52D7  A901        L52 LDA #1
52D9  8D1006       STA HUM
52DC  4CDF52       JMP WAK

                                            ;POKE MOVEMENT
52DF  AD2606      WAK LDA EXPL          ;VERTICAL MOVEMENT
52E2  C901         CMP #1
52E4  D003 ^52E9   BNE L421
52E6  4C8853       JMP EXP
52E9  CE1406      L421 DEC COUNT
52EC  AD1406       LDA COUNT
52EF  CD0506       CMP UMOVE
52F2  F008 ^52FC   BEQ L60
52F4  CD0606       CMP DMOVE
52F7  F00E ^5307   BEQ L61
52F9  4C1253       JMP WAR

52FC  A9FF        L60 LDA #255              ;UP
52FE  8D1406       STA COUNT
```

Assembly Listing

```
ATARI Macro Assembler Ver  1.0A  Page  11
                                  D1:LANDER.

5301  CE0306          DEC VERT
5304  4C1253          JMP WAR

5307  A9FF       L61  LDA #255       ;DOWN
5309  8D1406          STA COUNT
530C  EE0306          INC VERT
530F  4C1253          JMP WAR

5312  EE1706     WAR  INC SLOW       ;HORIZONTAL MOVEMENT
5315  AD1706          LDA SLOW       ;SLOW TURNS IN DIRECTION
5318  C905            CMP #5
531A  F003 ^531F      BEQ L200
531C  4C4D53          JMP COL
531F  A900      L200  LDA #0
5321  8D1706          STA SLOW

5324  CE1506          DEC COUNT1     ;SLOWER
5327  AD1506          LDA COUNT1
532A  CD1306          CMP LMOVE      ;HORZ
532D  F008 ^5337      BEQ L62
532F  CD1206          CMP RMOVE
5332  F00E ^5342      BEQ L63
5334  4C4D53          JMP COL

5337  A9FF       L62  LDA #255       ;LEFT
5339  8D1506          STA COUNT1
533C  CE0206          DEC HORZ
533F  4C4D53          JMP COL

5342  A9FF       L63  LDA #255       ;RIGHT
5344  8D1506          STA COUNT1
5347  EE0206          INC HORZ
534A  4C4D53          JMP COL

534D  AD0406     COL  LDA EXPL          ;COLLISON SECTION
5350  C900            CMP #0  -     ;EXPLOSION STARTED
5352  F003 ^5357      BEQ L450
5354  4C8B53          JMP EXP           ;GOTO EXPLOSION
5357  AD04D0    L450  LDA $D004         ;COLLISON WITH PLAYFIELD
535A  C900            CMP #0
535C  D02D ^538B      BNE EXP
535E  AD08D0          LDA $D008      ;COLLISIONS WITH MISSILE(0)
5361  C900            CMP #0
5363  D026 ^538B      BNE EXP
5365  AD0AD0          LDA $D00A      ;COLLISION WITH MISSILE(2)
5368  C900            CMP #0
536A  D01F ^538B      BNE EXP
536C  AD0BD0          LDA $D00B      ;COLLISION WITH MISSILE(3)
536F  C900            CMP #0
5371  D018 ^538B      BNE EXP
5373  AD0ED0          LDA $D00E      ;COLLISION WITH PLAYER(2)
5376  C900            CMP #0
5378  D011 ^538B      BNE EXP
537A  AD0CD0          LDA $D00C      ;HIT LANDING PAD
537D  C908            CMP #8
537F  F007 ^5388      BEQ PT
5381  C90A            CMP #10
```

```
ATARI Macro Assembler Ver  1.0A  Page  12
                                  D1:LANDER.

5383  F003 ^5388      BEQ PT
5385  4CE14E          JMP DRAW

5388  4C2353     PT   JMP PTS        ;JUMP TO SCORE SECTION

538B  A900       EXP  LDA #0            ;EXPLOSION SECTION
538D  8DC102          STA $2C1          ;ERASE THRUST FLAMES
5390  A901            LDA #1
5392  8D0406          STA EXPL       ;LOAD EXPLOSION FLAG
5395  AD2306          LDA ECON1      ;CHECK EXPLOSION LOOP
5398  C90A            CMP #10
539A  D001 ^539D      BNE L405
539C  60              RTS
539D  EE2706     L405 INC EXCON         ;EXPLOSION COUNTER
53A0  AE2706          LDX EXCON      ;EXPLOSION SLOWER
53A3  E005            CPX #5
53A5  F003 ^53AA      BEQ L300
53A7  4CE14E          JMP DRAW
53AA  A200       L300 LDX #0         ;RESET EXPLOSION SLOWER
53AC  8E2706          STX EXCON
53AF  C900            CMP #0
53B1  F003 ^53B6      BEQ L406
53B3  4CC354          JMP A5
53B6  EE2406     L406 INC ECON       ;INCREMENT EXPLOSION TYPE
53B9  AD2406          LDA ECON       ;LOAD EXPLOSION TYPE
53BC  C901            CMP #1         ;CHECK TYPE 1
53BE  D003 ^53C3      BNE L399
53C0  4CE253          JMP A1
53C3  C94B       L399 CMP #75        ;CHECK TYPE 2
53C5  D003 ^53CA      BNE L400
53C7  4C2354          JMP A2
53CA  C996       L400 CMP #150       ;CHECK TYPE 3
53CC  D003 ^53D1      BNE L401
53CE  4C3954          JMP A3
53D1  C9AF       L401 CMP #175       ;CHECK TYPE 4
53D3  D003 ^53D8      BNE L402
53D5  4C8054          JMP A4
53D8  C9FA       L402 CMP #250       ;CHECK TYPE 5
53DA  D003 ^53DF      BNE L403
53DC  4CC354          JMP A5
53DF  4CE14E     L403 JMP DRAW

53E2  A9DE       A1   LDA #222          ;TYPE 1
53E4  8DC002          STA $2C0           ;COLOR
53E7  AC0306          LDY VERT          ;VERTICAL LOCATION
53EA  A900            LDA #0          ;DRAW
53EC  99FE7B          STA PM-2,Y
53EF  99097C          STA PM+9,Y
53F2  99FF7B          STA PM-1,Y
53F5  99007C          STA PM,Y
53F8  99087C          STA PM+8,Y
53FB  99077C          STA PM+7,Y
53FE  99017C          STA PM+1,Y
5401  A92A            LDA #42
5403  99027C          STA PM+2,Y
5406  99067C          STA PM+6,Y
5409  A91C            LDA #28
```



```
ATARI Macro Assembler Ver  1.0A  Page  13
                                          D1:LANDER.

5408  99037C      STA PM+3,Y
540E  99057C      STA PM+5,Y
5411  A97E        LDA #126
5413  99047C      STA PM+4,Y
5416  A918        LDA #24
5418  8D00D2      STA $D200          ;SOUND
541B  A90E        LDA #14
541D  8D01D2      STA $D201          ;SOUND
5420  4CE14E      JMP DRAW

5423  A90B      A2 LDA #11            ;TYPE 2
5425  8DC002      STA $2C0           ;COLOR
5428  AC0306      LDY VERT          ;VERTICAL LOCATION
542B  A92A        LDA #42             ;DRAW
542D  99017C      STA PM+1,Y
5430  A949        LDA #73
5432  99027C      STA PM+2,Y
5435  A91C        LDA #28
5437  99037C      STA PM+3,Y
543A  99057C      STA PM+5,Y
543D  A9F7        LDA #247
543F  99047C      STA PM+4,Y
5442  A908        LDA #8
5444  99067C      STA PM+6,Y
5447  A922        LDA #34
5449  99077C      STA PM+7,Y
544C  A920        LDA #32
544E  8D00D2      STA $D200         ;SOUND
5451  A90E        LDA #14
5453  8D01D2      STA $D201         ;SOUND
5456  4CE14E      JMP DRAW

5459  A935      A3 LDA #53            ;TYPE 3
545B  8DC002      STA $2C0             ;COLOR
545E  AC0306      LDY VERT             ;VERTICAL LOCATION
5461  A918        LDA #24             ;DRAW
5463  99017C      STA PM+1,Y
5466  99067C      STA PM+6,Y
5469  A924        LDA #36
546B  99027C      STA PM+2,Y
546E  99057C      STA PM+5,Y
5471  A943        LDA #67
5473  99037C      STA PM+3,Y
5476  A9C2        LDA #194
5478  99047C      STA PM+4,Y
547B  A908        LDA #8
547D  99077C      STA PM+7,Y
5480  A964        LDA #100
5482  8D00D2      STA $D200           ;SOUND
5485  A905        LDA #5
5487  8D01D2      STA $D201           ;SOUND
548A  4CE14E      JMP DRAW

548D  A938      A4 LDA #56            ;TYPE 4
548F  8DC002      STA $2C0            ;COLOR
5492  AC0306      LDY VERT            ;VERTICAL LOCATION
5495  A942        LDA #66             ;DRAW
```

```
ATARI Macro Assembler Ver  1.0A  Page  14
                                          D1:LANDER.

5497  99007C      STA PM,Y
549A  99077C      STA PM+7,Y
549D  A981        LDA #129
549F  99017C      STA PM+1,Y
54A2  99067C      STA PM+6,Y
54A5  A900        LDA #0
54A7  99027C      STA PM+2,Y
54AA  99037C      STA PM+3,Y
54AD  99047C      STA PM+4,Y
54B0  99057C      STA PM+5,Y
54B3  99087C      STA PM+8,Y
54B6  A93C        LDA #60
54B8  8D00D2      STA $D200              ;SOUND
54BB  A90A        LDA #10
54BD  8D01D2      STA $D201              ;SOUND
54C0  4CE14E      JMP DRAW

54C3  AC0306     A5 LDY VERT             ;TYPE 5
54C6  A900        LDA #0             ;ERASE EXPLOSION
54C8  99FE7B      STA PM-2,Y
54CB  99FF7B      STA PM-1,Y
54CE  99007C      STA PM,Y
54D1  99017C      STA PM+1,Y
54D4  99027C      STA PM+2,Y
54D7  99037C      STA PM+3,Y
54DA  99047C      STA PM+4,Y
54DD  99057C      STA PM+5,Y
54E0  99067C      STA PM+6,Y
54E3  99077C      STA PM+7,Y
54E6  99087C      STA PM+8,Y
54E9  99037B      STA M1+3,Y         ;ERASE SIDE THRUST
54EC  99047B      STA M1+4,Y
54EF  99057B      STA M1+5,Y
54F2  99057D      STA PM1+5,Y        ;ERASE BOTTOM THRUST
54F5  99067D      STA PM1+6,Y
54F8  99077D      STA PM1+7,Y
54FB  99087D      STA PM1+8,Y
54FE  99097D      STA PM1+9,Y
5501  990A7D      STA PM1+10,Y
5504  990B7D      STA PM1+11,Y
5507  990C7D      STA PM1+12,Y
550A  990D7D      STA PM1+13,Y
550D  A900        LDA #0              ;NO SOUND
550F  8D00D2      STA $D200
5512  8D01D2      STA $D201
5515  EE2506      INC ECON1          ;INCREMENT EXPLOSION LOOP
5518  AD2B06      LDA SCON          ;LOAD SCORE LOOP
551B  C900        CMP #0
551D  F001 ^5520  BEQ J8
551F  60          RTS               ;IF SCORE LOOP IS 1  THEN END
5520  4CE14E     J8 JMP DRAW

5523  AD0606     PTS LDA DMOVE      ;SCORE SECTION
5526  38          SEC
5527  ED2F06      SBC STREN         ;CHECK    LANDER DOES NOT HIT TOO HARD
552A  3003 ^552F  BMI J57
552C  4CB853      JMP EXP           ;GOTO EXPLOSION
```

ADVERTISERS

ABBS..................................48

THE ALIEN GROUP..........IFC

EASTERN HOUSE..............29

FRONTRUNNER...................4

MICROPROSE....................12

MIGHTY BYTE..................27

MILES COMPUTING.........OBC

COMPUTER PALACE.......IBC

```
                         ATARI Macro Assembler Ver  1.0A  Page  15
                                                         D1:LANDER.

5552F  AD2B06       J57 LDA SCON        ;SCORE LOOP COUNTER
5532   C900          CMP #0             ;IF 0 THEN DETERMINE SCORE
5534   F00D ^5543    BEQ J6
5536   AD1F00        LDA CONSOL    ;CHECK IF START BUTTON IS PRESSED
5539   C906          CMP #6            ;YES: CLEAR LANDER DATA
553B   D003 ^5540    BNE J7            ;NO: CONTINUE
553D   4C0354        JMP A5
5540   4CE14E       J7 JMP DRAW
5543   A901         J6 LDA #1             ;SET SCORE LOOP
5545   8D2B06        STA SCON
5548   A9FF          LDA #255               ;SCORE SECTION
554A   38            SEC                      ;DOWN MOVEMENT
554B   ED0A06        SBC DMOVE
554E   8D2A06        STA SCORE
5551   A9FF          LDA #255
5553   38            SEC                      ;RIGHT MOVEMENT
5554   ED1206        SBC RMOVE
5557   8D2906        STA SCORE1
555A   A9FF          LDA #255
555C   38            SEC                      ;LEFT MOVEMENT
555D   ED1306        SBC LMOVE
5560   8D2C06        STA SCORE3

5563   AD0206        LDA HORZ           ;MEASURE CLOSENESS OF LANDING
5566   38            SEC
5567   ED9106        SBC #691
556A   C900          CMP #0
556C   D008 ^5576    BNE J51
556E   A9FF          LDA #255           ;PIN POINT LANDING
5570   8D2E06        STA PIN
5573   4C8155        JMP J50
5576   8D2D06       J51 STA STORE
5579   3006 ^5581    BMI J50
557B   A9FF          LDA #255
557D   38            SEC
557E   ED2D06        SBC STORE
5581   8D2A06       J50 STA SCORE2      ;STORE VALUE

5584   A900          LDA #0                 ;ERASE THRUST
5586   AC0306        LDY VERT
5589   99077D        STA PM1+7,Y
558C   99087D        STA PM1+8,Y
558F   99097D        STA PM1+9,Y
5592   990A7D        STA PM1+10,Y
5595   990B7D        STA PM1+11,Y
5598   990C7D        STA PM1+12,Y
559B   990D7D        STA PM1+13,Y
559E   8D00D2        STA $D200              ;NO SOUND
55A1   8D01D2        STA $D201
55A4   99037B        STA M1+3,Y             ;ERASE SIDE THRUST
55A7   99047B        STA M1+4,Y
55AA   4CE14E        JMP DRAW


no ERRORs,  140 Labels, $244A free.
```

COMPUTER PALACE

Formerly Royal Software

Make Computer Palace Your ATARI Source!!!

ATARI

Over 1000 items for your Atari!

Multiple Screens
Great Graphics
Action-Adventure
Multiple Players

OUR PRICE **$31.50**
Reg $34.95

An interesting variation on the Climb & Jump game!

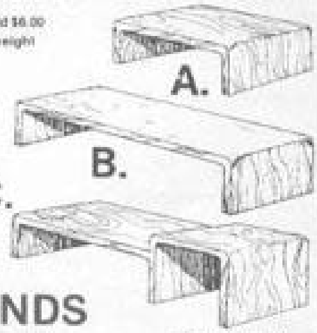
Pharaoh's Curse

From SYNAPSE — 32K Tape, Disk

Sits over computer to hold your TV/Monitor.

Please add $6.00 each for freight

Solid Oak



T.V. STANDS

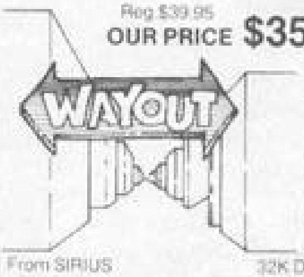
A. 800/400 + TV
B. 800/400 + 810 + TV
C. 800/400 + 2-Percoms + TV

- *Hand-made*
- *Beautiful finish*

A. = $29.95 B. = $39.95 C. = $49.95

- Fantastic 3-D Maze Game
- Continuous 3-D scrolling!
- 26 different mazes!!!
- Save-Game feature, too!

Reg $39.95
OUR PRICE **$35.10**



"CAN YOU FIND YOUR WAY-OUT?!"

From SIRIUS — 32K Disk Only

FOOT BALL

★ Pass ★ Run ★ Kick

- One or Two Players
- 24K RAM Disk or Tape

Great Graphics
Scrolling Field

Reg $31.95
OUR PRICE
$28.90

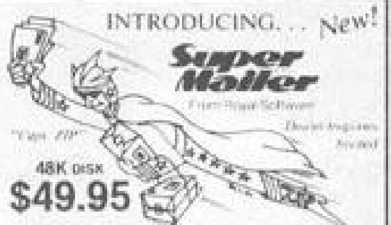
From Gamestar

INTRODUCING. . . *New!*

Super Mailer

From Royal Software

Dealer Inquiries Invited



48K DISK
$49.95

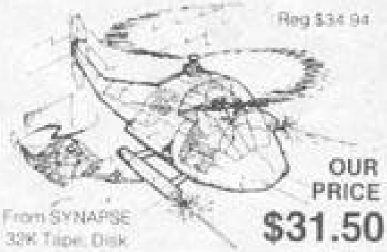
The most user-friendly mail maintenance program available! Here are a few of the outstanding features. . .

- New OS that offers fewer 'crashed-disks'-True random access-Store 500 to 2000 names/addresses per disk
- Works on single or double density
- Automatic delete of duplicates
- Print a disk directory
- Official state abbreviations are built-in
- Make back-up copies ● Merge files
- Search files ● Sort files
- Create sub-files ● Print lables/file copy

FORT APOCALYPSE

- Mulitple Screens
- Vertical & Horizontal Scrolling!
- Excellent graphics—Exciting!

Reg $34.94



From SYNAPSE
32K Tape, Disk

OUR PRICE **$31.50**

TRI-PAC I

A very handy disk utility!!!

New! *From Royal Software* — Dealer Inquiries Invited

- Tape-To-Disk copy
- Screen Dump
- Disk Speed Check

Back-up those troublesome and slow-to-load cassette programs—Dump screen text to printer—Check the speed of your disk drive!

16K Disk **OUR PRICE $29.95**

Protect your valuable diskettes. This storage system holds 10 diskettes, and opens up to an easel-back position for easy access to programs.

Our Price ONLY $4.95

- Folds up for storage
- Made of tough plastic

FLIP PAC™

From Big Five

MINER 2049er

- 10 Different Screens
- 16K Cartridge
- Spectacular Sound & Graphics

Reg $49.95

OUR PRICE ONLY **$44.90**

DUST COVERS

- 400/800 Computer
- 410 Recorder
- 810 Disk Drive

Only $8.95 each
Please Specify

COMPUTER PALACE

Formerly Royal Software

OPEN Monday - Saturday, 9 to 6

2160 W. 11th Avenue Eugene, Oregon 97402

Credit Card Orders ONLY:
TOLL-FREE
1-800-452-8013
MasterCard VISA American Express

Free Catalog!

With any order.
Or you can send $1.00
(Refundable with purchase).
Send us your name & address for our FREE FLYERS.

We specialize in *ATARI Compatible Products*
So we know what works best!!!
Call for the latest info. . . (503)683-6620

Shipping: UPS Parcel Post + $2.00 — UPS Air + $[illegible]
Shipping on hardware & paper items Call for rates

24-48 Hour Shipping Available...CALL

ATARI®

PAYROLL SOFTWARE for the ATARI 400/800/1200XL

Miles Payroll System™ is an advanced and comprehensive payroll accounting system designed for businesses today. Cumulative totals are maintained for each employee, as well as complete reporting, check writing, and W-2 reporting. Some features included:

- Random access file organization for fast updating of individual records.
- Allows weekly, biweely, semimonthly or monthly pay periods.
- Completely menu-driven and user-friendly.
- Regular, Overtime, Double time, Sick, Holiday, Vacation, Bonus and Commission earning categories.
- Payroll deductions include Federal W/H Tax State W/H Tax, City W/H Tax, FICA, State Disability Insurance, Group Insurance and 3 user-defined deductions.
- Tax sheltered annuity deduction capability for IRAs and other tax shelters.
- State and Federal Unemployment Insurance maintained.
- Complete file viewing and editing capability.
- Maintains up to 50 employees per data diskette.
- Up to 10 user-defined Worker's Compensation classifications.
- Federal Tax tables may be changed in only 15 minutes each year by user when IRS changes tax.
- Table method used for State and City Tax, allowing compatibility with any state's or city's tax.
- Produces 15 different reports, including W-2 Forms Report.
- Checks calculated and printed automatically.
- 3 user-defined payroll deductions to accommodate customized needs such as savings, profit sharing, tax shelters, pensions, etc.
- Pay period, monthly, quarterly and yearly cumulative totals maintained for each employee.
- Automatic input error detection and recovery protects system from user-generated errors.
- Easy-to-follow, detailed, and comprehensive user's manual and tutorial leads the user step by step allowing anyone with little computer experience to easily operate the package. Includes index.
- Color, sound, and graphics utilized for user ease.
- Maintains employee pay history.
- Allows for manual payroll check writing.
- Packaged in a handsome 3-ring deluxe pocketed binder with 3 diskettes and manual.
- Reasonable price.

Price $179.95. Requires 32K and two Atari 810 disk drives. Payment in U.S. funds required with order.
California residents add 6.5% sales tax. C.O.D. or prepayment only.

COMING SOON

- Order/Entry Invoicing for stand alone operation and/or automatic posting to Atari's Bookkeeper System.
- Program to post MILES PAYROLL SYSTEM payroll data to Atari's Bookkeeper System.
- Program to create VisiCalc™ files from MILES PAYROLL SYSTEM payroll data.

See your local dealer, or contact Miles Computing.

Miles™

MILES COMPUTING
7136 Haskell Ave. #204
Van Nuys, CA 91406
(213) 994-6279

VisiCalc is a trademark of VisiCorp.
Atari is a trademark of Atari, Inc., a Warner Communications Co.
Miles Computing and MILES PAYROLL SYSTEM are trademarks of Miles Computing, Van Nuys, California. Not affiliated with Atari, Inc.

Dealer and Distributor inquiries welcome.