

LUNAR LANDER



© 1983 SELECCIONES ILUSTRADAS ARTIST: V. SEGRELLES

Welcome to *Lunar Lander*, a game for the Commodore 64 in which you control a lunar module attempting to successfully land on the moon's surface.

While playing, you will notice that there are two levels, or "screens."

In the first level, you must touch down on one of the landing pads in the mountains. You have the choice of landing at either the 500 point pad or the 100 point pad. Once you have landed at both pads, however, the 100 point pad disappears, leaving a pit in its place.

The second level is reached by navigating the Lander down into the pit. In this level, your goal is to land at the 1000 point pad in the cave—a task as hard as it looks.

You can control the lander with either a joystick or keyboard. If you use a joystick, be sure it is plugged in port 2. The fire button fires your retro rockets, and left or right movement is accomplished by moving your joystick left or right. With the keyboard, the space bar fires your retro rockets and the two cursor control keys at the bottom right of

the C-64 keyboard control left and right movement. If you use the keyboard, the retro rockets cannot be fired at the same time you are thrusting left or right. This limitation, however, does not apply to the joystick.

When the game is first run, you will be asked to select the gravity strength (1, 2, or 3). One is the easiest and three is almost impossible.

After you land or crash, press the "f7" function key to continue play.

THE PROGRAM

Lunar Lander is written in BASIC, except for a small machine language subroutine. The mountains are PRINTed onto the screen with the standard graphic characters found on the keyboard, while the lander and explosions are sprites.

While I'm not going to explain how the whole program works, I will cover some points that can be used in your own programs. The lander is actually three separate sprites: the lander with no flame, the lander with a small flame, and the

Continued on page 76

LUNAR LANDER

Continued from page 35

lander with a large flame. Flickering of the flame is accomplished by switching back and forth between different size flames, or between a flame and no flame.

A small machine language subroutine is used to move the sprites around in memory, but it could be used for other purposes. The actual subroutine is a block move; it moves from 1 to 256 bytes from any source address to any destination address in memory.

The routine is 14 bytes long, and located at line 2390. To use it, first poke it into protected memory somewhere (for example, the cassette buffer), and then call it with a SYS command. The source address, destination address, and number of bytes to move can be changed by changing the appropriate DATA values according to this table:

	relative byte(s)	my value(s)
source	4 and 5	8,158
destination	7 and 8	64,3
byte count	11	63

Notice that the source and destination are two numbers—it takes two bytes to store an address. The address is given with the high byte first, then the low byte, and can be calculated by this formula: high byte=INT(addr/2556), low byte=addr-high byte X 256.

Another use for the block move subroutine is to fill memory locations. For example, to clear the first 255 bytes of the screen, you set the destination address to 1025, the source address to 1024, and the byte count to 255. Now poke 1024 with 32 and call the block move subroutine. The routine will copy the value from 1024 to 1025, then from 1025 to 1026, etc., filling 255 bytes in this manner.

Finally, in the *Lunar Lander* program, you can change gravity to any value you want by changing GR in lines 715-730. Good luck and have fun! □
SEE PROGRAM LISTING ON PAGE 83



WHO PROGRAMMED J.R.

Continued from page 50

You must, therefore, avoid J.R. at all costs: first, because he will want to stop Sue Ellen from obtaining this independence, and second, because he will most assuredly feel that the map is rightfully his. J.R. will use any means to stop you!

"Sue Ellen goes on to say that you will be paid handsomely for your services and, if you find the map and return it safely to her, Sue Ellen will pay you a bonus of two million dollars.

"Unbeknownst to you and Sue Ellen, J.R. listens to your conversation while lurking in the hallway outside the living room doors.

"You must leave at once. She gives you the snapshot, the ring, and \$500,000 for expenses. She leaves you there and proceeds upstairs to her room."

According to Garon, the game has recently been sent out for play-testing, and no one has solved it yet. He can solve it in about 100 moves, but estimates that if the average person played for a few hours each night, finishing the game would take a couple of weeks "at the very least."

Which is really nothing to complain about. The good guys on the show have been trying to beat J.R. for years. □

EASY SCRIPT

Continued from page 17

output format. The latter is controlled entirely by a very complete set of embedded format commands. These include control of headers, footers, separate margins for headers, footers and text, page length, centering, top and bottom margins, forced paging, single, double, and triple line spacing, left and right justification and automatic page numbering. About the only feature lacking is the ability to do footnotes. (Note the distinction between footers and footnotes.) Printing widths of up to 240 columns are supported. Remember that the width of the output text is not the same as the width of the displayed text. Format commands are all preceded by hitting the f3 function key which displays as a re-

LUNAR LANDER

From page 76

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```

•5 REM DEC. 19, 1983
•6 POKE 53280,8:POKE53281,0
•10 PRINT"{SC}{WH}{CD}{CD}{CD}
      LUNAR LANDER"
•20 PRINT"{CD}          (C) COPYRI
      GHT 1983"
•30 PRINT"          BY ANTHONY W
      OOD"
•50 POKE55,248:POKE56,157:REM 4044
      0
•60 CLR:S=54272
•100 V=53248:S1=64*13:S2=64*14:S3=
      64*15:K=2040:SR=40444
•110 FORZ=0TO39:READA:POKE40456+Z,
      A
•120 POKES2+Z,A:POKES3+Z,A:NEXT
•130 FOR Z=40TO62:READA:POKE40456+
      Z,A:NEXT
•132 FORZ=40TO62:READA:POKES2+Z,A:
      NEXT
•134 FORZ=40TO62:READA:POKES3+Z,A:
      NEXT
•135 FORZ=1TO63*3:READA:POKE40518+
      Z,A:NEXT
•137 FORZ=0TO13:READA:POKE40441+Z,
      A:NEXT
•138 SYS40441
•139 FU=5000:F2=0:L1=0:L2=0:SC=0:G
      OSUB700
•140 GOSUB900:POKEV+39,1:POKEK,13:
      POKEV,0:POKEV+1,0:POKEV+21,1
•150 H=15:E=5:X=30+INT(RND(0)*100)
      :Y=30:F(0)=13:F(1)=13:F(2)=14
•160 F(3)=15:M=1
•200 J=PEEK(56320):P=PEEK(197):IFF
      U=0THENB=0:M=1:POKES+1,0:GOTO220
•210 IFP=60OR(JAND16)=0THENB=B+1:G
      OTO213
•211 B=0:M=1:POKES+4,0
•213 IFM<>1THENPOKES+4,129
•215 IF P=7OR(JAND4)=0THENH=H-1:PR
      INT"{HM}{CD}{CR}<<":FU=FU-1
•216 IF P=2OR(JAND8)=0THENH=H+1:PR
      INT"{HM}{CD}{CR}>>":FU=FU-1
•220 GOSUB760:IFB=6ORB=1THENM=M+1
•240 E=E+GR-M*1.65:FU=FU-(M-1)*10:
      IFFU<0THENFU=0
•245 X=X+INT(H/6):Y=Y+INT(E/5)
•250 IFX<5THENX=5
•252 IFY>251THENY=252:GOSUB800:Y=3
      5:X=29:FU=5000:F2=1:GOSUB5000
•255 IFY<35THENY=35:E=3
•256 IFX>319THENX=319
•260 POKEK,13:Z=PEEK(V+31):GOSUB80
      0
•266 IFPEEK(V+31)<>0THEN300
•270 POKEK,F(M-FL):IFFL=0THENFL=1:
      GOTO200
•280 FL=0:GOTO200
•300 IFF2=1THEN440
•310 IFY<149THEN270
•320 IFY<227 ORX<099OR X>123 ORE>9
      ORABS(H)>10THEN330
•322 IF L1=1ANDL2=1THEN330
•325 Y=227:GOSUB800:SC=SC+100:L2=1
•327 GOTO400
•330 IFY<177 ORX<197 ORX>203 ORE>9
      ORABS(H)>10THEN340
•335 Y=179:GOSUB800:SC=SC+500:L1=1
      :GOTO400
•340 POKES+4,129:V2=11:POKESR,63+8
      :SYS40441:GOSUB820:POKESR,126+8:S
      YS40441
•350 GOSUB840:POKESR,189+8:SYS4044
      1:GOSUB840:POKES+4,0:POKES+24,15:
      POKES+1,2
•360 POKEV+21,0:POKESR,8:SYS40441:
      GOTO410
•400 POKES+4,0:GOSUB750:PRINT"{HM}
      LANDED!!"
•410 GETA$:IFA$<>"{F7}"THEN410
•415 POKEV+21,0:PRINT"{HM}
      "
•420 IFFU=0THEN 460
•425 IFF2=1THENY=10:X=29:E=5:H=0:G
      OSUB5000:POKEV,X:POKEV+1,Y:POKEV+
      21,1:GOTO160
•435 GOTO140
•440 IFX<230ORX>242ORY<121ORY>130O
      RE>90RABS(H)>11THEN340
•450 F2=0:SC=SC+1000:L1=0:L2=0:FU=
      500:GOTO400
•460 PRINT"{SC}":PRINT:PRINT"DO YO

```



```

U WANT TO PLAY AGAIN?"
•470 GETA$:IFA$=""THEN470
•480 IFA$="Y"THEN139
•490 IFA$<>"N"THEN470
•500 POKES+4,0:PRINT"{SC}":END
•700 PRINT:PRINT:PRINT" ENTER GRA
VITY STRENGTH ({RV}1{RO},{RV}2{RO
}, OR {RV}3{RO})"
•701 POKES+5,129:POKES+6,216:POKES
+24,15
•702 POKES,200:POKES+4,0:POKES+1,2
•710 GETA$:IFA$=""THEN710
•715 IFA$="1"THENGRA=3:RETURN
•720 IFA$="2"THENGRA=3.6:RETURN
•730 IFA$="3"THENGRA=4.25:RETURN
•740 GOTO710
•750 PRINT"{HM}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}
{CR}{CR}{CR}{CR}SCORE";SC;
•760 PRINT"{HM}{CD}{CR}{HM}";TAB
(28);"FUEL {CL}{CL}{CL}{CL}{C
L}";FU;:RETURN
•800 POKEV+1,Y:IFX<256THENPOKEV,X:
POKEV+16,0:RETURN
•810 POKEV,X-256:POKE53264,127:RET
URN
•820 POKES+1,3:FORX=1TO4:POKES+24,
V2:FORV=1TO15:NEXT:V2=V2+1
•830 NEXT:RETURN
•840 FORX=1TO7:POKES+24,V2:FORV=1T
O9:NEXT:V2=V2-1:NEXT:RETURN
•900 PRINT"{SC}":GOSUB750:PRINT:FO
RY=0TO9:A=RND(0)
•910 IFA>.3THENPRINTTAB(RND(0)*39)
;".
•920 IF A<=.3THENPRINT
•930 NEXT:PRINT:PRINT:PRINT
•1000 PRINT"
{RV}⌊⌋{RO} "
•1100 PRINT"
{RV}⌊⌋ {RO} "
•1200 PRINT"
{RV}⌊⌋ {RO} "
•1300 PRINT"
{RV}⌊⌋ {RO} "
•1400 PRINT"
RV⌊⌋ {RO} "
•1500 PRINT"
V⌊⌋ 500 {RO}"
•1600 PRINT"
{RV}⌊⌋ {RO} {RO} "
•1700 PRINT"
{RV}⌊⌋ {RO} {RO}
"
•1800 PRINT" {RV}⌊⌋ {RO}
{RV}⌊⌋ {RO}
"
•1900 PRINT" {RV}⌊⌋ {RO}
{RV}⌊⌋ {RO}
"
•1910 PRINT"{RV} 100
{RO}";
•1920 IFL2=1ANDL1=1THENPRINT"{HM}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CR}
{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{
CR}{RV}⌊⌋{RO} {RV}⌊⌋{RO}";
•1930 RETURN
•1999 REM LANDER SPRITES
•2000 DATA 0,255,0,0,255,0,7,255,2
24,7
•2010 DATA 255,224,7,255,224,0,255
,0,0,255
•2020 DATA 0,15,255,240,15,255,240
,15,255,240
•2030 DATA 3,255,192,3,60,192,6,12
6,96,12
•2040 REM NO FLAME
•2050 DATA 0,48,24,0,24,0,0,0,0,0
•2060 DATA 0,0,0,0,0,0,0,0,0,0,0
,0
•2070 REM SMALL FLAME
•2080 DATA 0,48,24,60,24,0,102,0,0
,60
•2090 DATA 0,0,24,0,0,24,0,0,0,0,0
,0,0
•2100 REM BIG FLAME
•2110 DATA 0,48,24,126,24,0,219,0,
0,219
•2120 DATA 0,0,102,0,0,60,0,0,24,0
,0,24,0
•2130 REM EXPLOSION
•2140 DATA 0,0,0,0,0,0,0,0,0
•2150 DATA 0,0,0,0,0,0,0,0,0
•2160 DATA 64,0,6,72,0,0,24,0
•2170 DATA 12,224,0,0,228,0,0,0
•2180 DATA 0,13,80,0,0,0,0,0
•2190 DATA 0,0,0,0,0,0,0,0,0,0,0
,0
{R •2200 DATA 0,0,0,0,0,0,0,0,0,0
•2210 REM FRAME 2 OF EXPLOSION
•2220 DATA 0,0,0,0,0,0,0,0,0
•2230 DATA 0,0,128,0,24,131,0,24
•2240 DATA 3,0,0,0,0,0,0,0
•2250 DATA 48,224,0,48,227,0,0,0

```


BUG REPELLENT LINE CODES FOR LUNAR LANDER

AHOY! 85

LINE # 400:GA
 LINE # 410:OI
 LINE # 415:OA
 LINE # 420:JA
 LINE # 425:KA
 LINE # 435:JA
 LINE # 440:DA
 LINE # 450:CA
 LINE # 460:EA
 LINE # 470:BI
 LINE # 480:GM
 LINE # 490:BI
 LINE # 500:FA
 LINE # 700:NA
 LINE # 701:PA
 LINE # 702:LI
 LINE # 710:BI
 LINE # 715:HI
 LINE # 720:OI
 LINE # 730:BI
 LINE # 740:NI
 LINE # 750:JE
 LINE # 760:FI
 LINE # 800:II
 LINE # 810:PI
 LINE # 820:HE
 LINE # 830:CI
 LINE # 840:CI
 LINE # 900:AE
 LINE # 910:DI
 LINE # 920:IM
 LINE # 930:EE
 LINE # 1000:KI
 LINE # 1100:KI
 LINE # 1200:KI
 LINE # 1300:KI
 LINE # 1400:OI
 LINE # 1500:II
 LINE # 1600:OI
 LINE # 1700:KI
 LINE # 1800:OI
 LINE # 1900:KI
 LINE # 1910:JM
 LINE # 1920:LM
 LINE # 1930:DI
 LINE # 1999:FE
 LINE # 2000:DM
 LINE # 2010:BM
 LINE # 2020:MA
 LINE # 2030:BA
 LINE # 2040:AM
 LINE # 2050:KA
 LINE # 2060:KA

LINE # 2070:IM
 LINE # 2080:DA
 LINE # 2090:KA
 LINE # 2100:AM
 LINE # 2110:AM
 LINE # 2120:KA
 LINE # 2130:GA
 LINE # 2140:KA
 LINE # 2150:KA
 LINE # 2160:KA
 LINE # 2170:KA
 LINE # 2180:KA
 LINE # 2190:KA
 LINE # 2200:KA
 LINE # 2210:GA
 LINE # 2220:KA
 LINE # 2230:KA
 LINE # 2240:KA
 LINE # 2250:KA
 LINE # 2260:CA
 LINE # 2270:CA
 LINE # 2280:KA
 LINE # 2290:GA
 LINE # 2300:KA
 LINE # 2310:KA
 LINE # 2320:KA
 LINE # 2330:KA
 LINE # 2340:CA
 LINE # 2350:MA
 LINE # 2360:KA
 LINE # 2370:KA
 LINE # 2380:JE
 LINE # 2390:MA
 LINE # 2400:AA
 LINE # 4999:HE
 LINE # 5000:LI
 LINE # 5005:FI
 LINE # 5010:FI
 LINE # 5020:FI
 LINE # 5030:FI
 LINE # 5040:FI
 LINE # 5050:KI
 LINE # 5060:LI
 LINE # 5065:LI
 LINE # 5070:LI
 LINE # 5080:JA
 LINE # 5090:LI
 LINE # 5100:PA
 LINE # 5200:DA
 LINE # 5210:DA
 LINE # 5220:DA
 LINE # 5230:GA
 LINE # 5240:KI

LINE # 5250:OI
 LINE # 5260:OI
 LINE # 5270:OI
 LINE # 5280:OI
 LINE # 5285:LI

LINE # 5290:LI
 LINE # 5295:FM
 LINE # 5300:HM
 LINE # 6000:DI
 LINES: 163

NAME THAT STAR

From page 78

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```

•1 REM ***** "NAME THAT STAR" V2.0
  *****
•2 REM **   DESIGNED BY: BOB LLORET
  **
•3 REM *****   FOR: AHOY' MAGAZINE
  *****
•4 REM
•60 PRINT "{SC}":POKE 53280,4:POKE
  53281,7
•70 PRINT TAB(8);"{BR}{CD}{CD}{CD}
  {CD}CONTESTANTS APPLICATION"
•80 PRINT TAB(8);"{BL}{CD}{CD}{CD}
  {CD}{CD}FIRST NAME:":INPUT NAMES$
•85 PRINT TAB(8);"{RD}{CD}{CD}{CD}
  {CD}CONGRATULATIONS...";NAME$;","
•90 PRINT TAB(8);"{RD}{CD}YOU HAVE
  BEEN ACCEPTED.":FOR D=1 TO 3000:
  NEXT D
•100 PRINT "{SC}":GOSUB 880:GOSUB
  900
•110 PRINT TAB(18);"{CU}{CU}{CU}{C
  U}{CU}{CU}{CU}";A$:PRINT TAB(18);
  "{CU}";A4$
•120 GOSUB 1000
•130 FOR D=1 TO 1500:NEXT D:PRINT
  "{G3}{CU}{CU}{CU}{CU}{CU}";
•140 GOSUB 1460
•170 FOR D=1 TO 3000:NEXT D
•180 PRINT "{CD}{CD}":GOSUB 1000:P
  RINT TAB(18);A$:PRINT TAB(18);A1$
•190 PRINT "{CU}{CU}{CU}{CU}{CU}{C
  U}{CU}{CU}":GOSUB 1460:PRINT TAB(
  16);"{CU}, ";NAME$
•200 FOR D=1 TO 2500:NEXT D:PRINT
  
```


Ahoy!

COMMODORE
USERS **K**

Ion International Inc. \$2.50/Can. \$2.75 April 1984

**PETSPPEED
-MAN'S
BEST
FRIEND!**

**LUNAR
LANDER
-A 64
CHALLENGER!**

**DALLAS
QUEST
ADVENTURE
GAME!**

EXCLUSIVE!

