Air Hockey

by Chris Page

Air Hockey is written in Action! and must be compiled off of a disk or tape (the source and object code won't fit in memory together). So type it in, SAVE it, clear the editor, go to the monitor and RUN it, thusly: R "FILENAME" (substituting the device and filename you saved it under for "FILENAME"—I used "D.AIROHOCKEY").

Once you have it running properly, you should see the title screen and hear the title music (the "Peter Gunn" bass line). Press START.

Now you should see the options screen (it has the word OPTIONS at the top). You can use the OPTION key to highlight a different option, SELECT to change the option and START to play the game.

The options available in Air Hockey are:

Friction — This can be ON or OFF. If the friction is on, the puck will have a tendency to slow down while travelling across the board. You may notice that it sometimes curves as it slows down. This is because I used integer values instead of floating point. This means that the motion is not 100 percent accurate, resulting in the curved motion of the puck.

Velocity — This can be 2 through 9 and indicates the maximum velocity of the puck. Option 2 is slowest; 9 is fastest.

Bounce — This can be 0 through 9, indicating the amount of "bounce" to the puck, or how well it retains velocity after hitting the side of the board. A 9 means that the puck will not slow down on collision; 0 indicates very little bounce and will cause the puck to slow down considerably if it hits the sides.

Win — This is the score up to which the player(s) will play. It can be from 20 to 90, in increments of 10.

Players — Either 1 or 2. If one player is selected, then the player should use joystick port 1 and control the top player; the computer will control player two, the bottom player. If two players are selected, then it's the same, except that player two will be controlled by joystick port 2.

Playing.

Once your options are set (or left alone, if you like the default settings) you may press START to play. You'll then see a vertical air hockey board with the scores displayed at the top, along with the score necessary to win the game. The puck will appear in front of the serving player's paddle. That player must hit the puck to start the game.

The game is something like Pong, with forward motion as well as side-to-side. Players control the paddles by moving the joystick in the direction they wish to move. The buttons do nothing. A score is made when the puck goes into the yellow goal area of a player, and the other player becomes the server.

If the puck gets stuck between players, as it can in real air hockey, you may re-serve by pressing the SPACE BAR. Also, while in the play mode, you may...
press the ESCape key to exit the program. Finally, if you want to restart the game, press START anytime during the game (except during the goal sequence), and you'll be returned to the options screen.

When the game is over, there's a long cheering sequence with whistling fans (if the crowd likes you) before you're returned to the title screen.

Why I did it or a tail of two ducks.

I was sitting around one day (I do that quite a lot) last summer, had just bought Action! and was becoming familiar with it. After reading some demos, I was ready to do something more substantial with it.

I figured that a good way to utilize Action!'s speed was to write some kind of fast-paced, arcade-type game. But I didn't want to write another space game or Pac-Man. I wanted to write something different and unique. Air Hockey may not be a completely unique game (it is similar to Pong), but it's different, and a change of pace from "Laser the Aliens!"

---

ATARI PRODUCTS from CAL COM INC.

Atari 520ST (mono & stereo)  CALL
Atari 520ST Software  CALL
Atari 520ST 1024K Upgrade (includes Bandic software) $200.00
Atari 130XE  $135.00
Atari 130XE + Omniview XE/XL  $185.00
Happy 1050 Disk Drive (Complete)  $309.95
Happy Enhancements 610/1050  $150.00
Omniview XE/XL  $49.95
Ramrod XE/XL  $68.00
U.S. Doubler  $56.00
O.S.S. Software (basic XE/XL, Mac 65, Action! etc.)  CALL
F.O.R.E.M. XE (855 program)  $65.00
Volksmodem 12 (300/1200 baud with Cable)  $199.95
Gemini 10X Printer  $199.00
LJK Letter & Data Perfect (49/80 columns)  $29.95
R-writer (Comes with software)  $39.95
P.R. Connection (Interface Module by JCD)  $65.00

5295 Cameron Drive #505 Silver Spring, MD 20902
(301) 651-9121

A lesson in compromise or the quacker in the rye.

This program is an example of inventiveness, procrastination and compromise. "Inventive" because... well, it's a matter of opinion, but I think it's inventive. "Procrastination" because I dropped the project for several months at a time (check out the start and completion dates in the source listing). Finally, and most decided, "compromise" because I made so many of these concerning Air Hockey.

Two of my main compromises were:

1) I wanted more options and a complete title song, but time and a willingness to work (or the lack thereof) got in the way.

2) Everyone, including myself, thought that the paddles should have been round. But that requires physics... I barely passed physics. If I'd made the paddles round, as they are in real life, I would have had to resort to "real" physics instead of the chintzy method I did use.

In physics, you use vectors to describe how objects move, but this is a difficult thing in integers (well, difficult for me), which is what I was stuck with in Action! So I simply gave a horizontal and vertical speed and a horizontal and vertical direction.

These two compromises, however, were not as difficult or as important as my final compromise. I had to compromise on the one aspect, the most difficult thing, that every programmer must: completion of my goal. I had to stop work on a program which I felt was incomplete and short of my goals, and call it finished.

I realized this when I was telling my friends that I'd have to add the treble line to the title music before I'd submit it for publishing. As I told them this, I realized that the game is rather simple (as it was supposed to be) and that a full-blown song was superfluous. I then realized that other things I wanted to add were also not needed.

Actually, I had already met my goal (design a simple game as an exercise in Action!), but in the process, I'd created other goals—like adding the treble line.

I realized that I would continue creating goals as long as I was working on it; I would never finish the game. I'd sit, perma-bonded to my video screen, for the next ten years working on Air Hockey until it was 3-D, talked and had instant replay, a high score list, a theme song to put "Flight of the Valkyries" to shame, and a thousand other things... and I would still want to change something.
Program design and some ducks thrown in for effect.

I think the important thing here is to realize that, when you want to write a program, you should decide exactly what it will be like, so that you can say it is finished when it meets the description. I certainly did not. I designed and wrote it as I went along (this is painfully evident to me in the lack of unity and consistency in the program, the "patchwork quilt" look).

This has also led to my big problem: because the program is so disorganized, I invariably come to some sort of dead end and drop the project. I completely gave up on Air Hockey many months back, but, at the urging of two of my friends (D.S. and D.B.), I picked it up again and trudged through the tangled code to finish it... finally. This is what has kept me from finishing the other hundred or so projects I have stored away in dusty disk files.

I'm sure that if it were not for this fact, there would be thousands more programs available for computers through other users, magazines and distributors. Next time you start to put something off because it seems too difficult, back up and try again.

Oh yes, the ducks.

The ducks? Well, I thought I'd try to be a little different from the other articles gracing this magazine's fine pages. (You wondered about them, didn't you?) Have a duck, you'll feel better. ☺

Chris Page is an eighteen-year-old from San Diego, who's studying for an A.A. degree in electronics at I.T.T. Technical Institute. He has worked with computers for seven years and owned an Atari 800 for four. His primary computing interest is in sound, graphics and human interfacing.

Listing 1.
Action! listing.

; =================
; = Air Hockey =
; by =
; = Chris Page =
; =================

; Copyright (c) 1985 ANALOG Computing

; Special Thanks To:
; David Sullivan & David Becker

DEFINE  
OPCITION='3', SELECT='5', START='6',  
NONE='2', LEFT='36', RIGHT='60',  
TOP='56',BOT='144'

BYTE MINDEX,VOLUME,FRICTION=[1],  
BOUNCE=[19],WIN=[18],PLAYERS=[2],

HUE,LUH,OPT,PUCKXD,PUCKYD,HITFLAG,  
SERVER,SEVERE,SERVERS:,SOMETHING=555,  
COLOR1=53279,CHACKT=755,MVSYNC=54262,  
VCOUNT=54263,CHR1=592,  
COLOR2=53270,CHR2=53271,  
COLOR3=53274,RTCLK=38,  
DRACTL=54272,LMARGN=83,RMARGN=83,  
CHBOS=756,PM4BASE=54279,  
HITCR=53277,6PRON=623,  
RANDOM=53770,6COL=53266,  
COLPML=53267,CH=764,ANNTP=106,  
ANDCTL=53768,ACTR=77,KEY

BYTE ARRAY  
DLIST,SCNMEM,AMFM,PMEM($800),  
BAR($0)=[$FF,$FF]  
POUN=[$B9,$B9,$B9,$B9,$B9,$B9,$B9,$B9]  
THDC=[$TT,$TT,$TT,$TT,$TT,$TT,$TT,$TT]  
NOTE=[$24,$23,$22,$21,$20,$22,$23,$24]  
YTOP=[$60,$60,$60,$60]  
SCORE=22,DLIST=560,5AVC=88,  
XITUBO=5462

; --- Miscellaneous Procedures ---

PROC SETUBO=$E45C(BYTE CMD,UBH1,  
UBI0)

PROC UBI()  
; UBI to play music  
SOUND(A,NOT(EINDEX),10,VOLUME)  
SOUND(1,NOTE(MINDEX)-2,10,VOLUME)  
VOLUME=2  
IF VOLUME=0 THEN  
VOLUME=$20  
MINDEX=$11  
IF MINDEX=$0 THEN  
MINDEX=0  
FI  
JMP XITUBO  
540 XITUBO
RETURN

PROC INITUBI()  
; Initialize music UBI  
MINDEX=15  
SNOIB()  
; Set deferred UBI vector  
SETUBO($7,UBI R5H 8,UBI)  
RETURN

PROC DEBounce()  
CARD  
; Debounce console keys  
FOR I=0 TO 5800 DO  
U8=CONS=NONE  
OD

ANALOG COMPUTING

JANUARY 1986 / PAGE 61
Air Hockey continued

; --- Title Screen ---

PROC INITTITLE
BYTE I
; Initialize title screen
GRAPHICS(0)
GPR0:17
GRCTL:0
SUBCTL:0
COLOR=1
HUE=0
DLIST=DLIST1
DO UNTIL VCOUNT=0
DO FOR I=5 TO 5 DO
DLIST(I+3)=DLIST(I)
DO SETBLOCK(DLIST,10,$70)
FOR I=13 TO 25 STEP 2 DO
DLIST(I)=0
DO SETBLOCK(DLIST+27,2,$70)
SETCOLOR(1,0,14)
SETCOLOR(2,0,0)
POSITION(11,0)
PRINT("Air Hockey")
POSITION(11,1)
PRINT("By: Chris Page")
POSITION(20,13)
PRINT("";D.S. and D.B.""")
POSITION(17,2)
PRINT("June 30, 1984 - August 5, 1985")
POSITION(7,4)
PRINT("Copyright (C) 1984")
POSITION(34,6)
PRINT("Press START")
SDMCTL=33
RETURN

PROC TITLECOLORS()
BYTE J
; Mid-screen color changes
HUE=42
IF HUE<2 THEN
J=163
FOR I=0 TO 30 DO
DO MSYNC=0
COLOR4=VCOUNT LSH 1+HUE
IF VCOUNT=48 THEN
COLOR4=0
DO UNTIL VCOUNT & 128
DO RETURN

PROC TITLE()
; Display title screen
INITTITLE()
INITVIC()
DO TITILECOLORS() UNTIL CONSL=START
DO SDMCTL=0
RETURN

; --- Game Options ---

PROC INITOPTIONS()
; Initialize procedure OPTIONS()
GRAPHICS(17)
SDMCTL=0
GRCTL=0
DO UNTIL VCOUNT=0
DO DEBOUNCE()
SCREEN=50COLORS DLIST=DLIST1
DLIST1=1
SETCOLOR(0,3,14)
SETCOLOR(2,0,14)
PRINTDE(6,"GAME OPTIONS")
POSITION(18,2)
PRINTD(6,"NEXT OPTION")
PRINTD(6,"SELECT")
POSITION(10,6)
PRINTD(6,"PLAY GAME")
POSITION(12,6)
PRINTD(6,"FRIC'TION")
IF FRIC'TION THEN
PRINTD(6,"N")
ELSE
PRINTD(6,"FF")
DO POSITON(18,0)
PRINTD(6,"VELOCITY")
PRINTD(6,"MAX")
POSITION(18,10)
PRINTD(6,"BOUNCE")
POSITION(18,12)
PRINTD(6,"WIN")
PRINTD(6,"PLAYERS")
PRINTD(6,"PLAYERS")
SDMCTL=34
OPT=0
RETURN

PROC OPTIONS COLORS(BYTE OPT)
; Mid-screen color changes
OPT=option line to hi-light
DO MSYNC=0
UNTIL VCOUNT=15
DO LUM=0
MSYNC=0
DO MSYNC=0
COLORL=LUM&0FXS20
LUM=42
UNTIL VCOUNT=25
DO MSYNC=0
COLOR4=0
COLOR4=6
DO MSYNC=0
UNTIL VCOUNT=40
DO COLOR4=2
OPT=0
LSH 3+41
MSYNC=0
UNTIL VCOUNT=OPT
DO COLOR4=0
DO MSYNC=0
UNTIL VCOUNT=OPT+8
CORO=258
RETURN

PROC OPTIONS()
CARD I
; GET GAME OPTIONS FROM PLAYER(S)
INITIPTIONS()
DO
FOR I=0 TO 10 DO
OPTIONCOLORS( OPTI
UNTIL CONSOL=START
OD
IF CONSOL=OPTION THEN
OPT=+1
IF OPT<5 THEN
OPT=0
FI
IF CONSOL=SELECT THEN
IF OPT=0 THEN
FRICTION=-11
IF FRICTION THEN
SCREMM(134)=46
SCREMM(135)=0
ELSE
SCREMM(134)=30
SCREMM(135)=30
FI
ELSEIF OPT=1 THEN
IF MAXU=300 THEN
SCREMM(173)==7
MAXU=200
ELSE
SCREMM(173)=+1
MAXU=+100
FI
ELSEIF OPT=2 THEN
IF BOUNCE=90 THEN
SCREMM(213)==9
BOUNCE=0
ELSE
SCREMM(213)==+1
BOUNCE+=+10
FI
ELSEIF OPT=3 THEN
IF KEYO=90 THEN
SCREMM(253)==-9
MINO=10
ELSE
SCREMM(253)=+1
MINO=+10
FI
ELSE
IF PLAYERS=2 THEN
SCREMM(253)==1
PLAYERS=1
ELSE
SCREMM(253)=+1
PLAYERS=2
FI
FI
UNTIL CONSOL=START
OD
SDMLT=61
FOR I=1 TO 7 DO
FOR J=0 TO 3000 DO OD
RAMFONT(538+1)=EDAT(I)
RAMFONT(539+1)=EDAT(I)
RAMFONT(648+1)=EDAT(I)
RAMFONT(656+1)=EDAT(I)
RAMFONT(664+1)=EDAT(I)
RAMFONT(672+1)=EDAT(I)
RAMFONT(728+1)=EDAT(I)
OD
RETURN

PROC POSPLAYER(CARD PLAYER
BYTE X,Y,LENGTH
BYTE ARRAY SHAPE)
; Position Player
HP05(PLAYER)=X
LEFT
PLAYER:=W$100+S$400
MOVEBLOCK(PWMEM+PLAYER+Y+TOP,
SHAPE,LENGTH)
RETURN

PROC POSPOLE BYTE PADDLE,X,Y)
; Position Paddle
POLOPLAY(PADDLE,X,Y,2,BAR)
RETURN

PROC POSPOUCK(CARD X,Y)
; Position PUCK
X:=W$100
Y:=S$100
POSPOUCK(2,X,Y,8,PUCK)
RETURN

PROC ERASEPADDLE(CARD PADDLE BYTE Y)
; Erase Paddle
PADDLE:=W$100+S$400
ZERO(PWMEM+PADDLE+Y+TOP,2)
RETURN

PROC ERASEPUCK(CARD Y)
; Erase PUCK
Y:=+100+TOP
ZERO(PWMEM+S$800+Y,8)
RETURN

PROC CLEARALL()
; Clear Player MEMORY
ERASEPADDLE(0,PADDLE)
ERASEPADDLE(1,PADDLE)
ERASEPADDLE(2,PADDLE)
RETURN

PROC INITGRAPH()
; Initialize PGM
PWMEM:=CARTOP(16):W$100
ZERO(PWMEM+S$800)

PAGE 64 / JANUARY 1986
PROC INITPLAY()
; Initialize game
GRAPHICS(0)
DRAWTABLE
DO UNTIL VCOUNT=0
DO SETUPV(Y$E4,$62)
START1
DEBOUNCE
INITPONG
SCREEN=SAVMSC
SCORE=0
SCORR=1
ostik=0
ostik=-15
SERVER=0
GAMEOVER=0
CSNSMC=1
DLIST=DLISTL
DLISTL2=DLISTL[3]+4
DLISTL3=DLISTL[4]
DLISTL4=DLISTL5
DLISTL5=S20
DLISTL6=S30
SETBLOCK(DLISTL,[8,21,4])
SETCOLOR(0,0,6)
SETCOLOR(1,0,14)
SETCOLOR(2,0,4)
SETCOLOR(3,2,14)
SETCOLOR(4,0,6)
POSITION(5,0)
PRINT("Hockey")
SAVMSC=16
POSITION(0,0)
PRINT("One : 00 | Win : X \| Two : 00", 0)
MOVEBLOCK(SCRMEM+55,TTOP(0))
FOR I=0 TO 663 STEP 32 DO
MOVEBLOCK(SCRMEM+1,TTOP(0))
DO MOVEBLOCK(SCRMEM+675,TTOP(0))
MAKEF(0)
SOUND(0,0,0,0,3)
KEY=0
PLS=OFF
RETURN

PROC MOVEPADDLE(BYTE PI)
BYTE STIK
; Move paddle
ERASEPDL(PDLX(P),PDLY(P))
STIK=STICK(P)
; Move puck 2 for one player game
IF PLAYERS=P THEN
STIK=5F
IF DDLX(Y)+2>(PPXXS10 THEN
STIK=-8
ELSE
STIK=-4
FI
IF DDLX(Y)-6>(PXXS10 THEN
STIK=-7
ELSEIF DDLX(Y)-8>(PXXS10 THEN
STIK=-1
ELSE
STIK=-2
IF RANDM(2) THEN
STIK=+1
FI
FI
FI
; Save stick position
OSTIK(P)=STIK
; Move paddle vertically
IF (STIK&0)=0 THEN
PDLY(P)=PDLY(Y)+1
IF PDLY(P)>(TTOP(0)) THEN
PDLY(P)=TTOP(0)
FI
ELSEIF (STIK&2)=0 THEN
PDLY(P)=PDLY(Y)-1
IF PDLY(P)<(TTOP(0)) THEN
PDLY(P)=TTOP(0)
FI
FI
; Move paddle horizontally
IF (STIK&A0)=0 THEN
PDLX(P)=PDLX(X)+1
IF PDLX(P)>(TTOP(0)) THEN
PDLX(P)=TTOP(0)
FI
ELSEIF (STIK&A0)=0 THEN
PDLX(P)=PDLX(X)-1
IF PDLX(P)<(TTOP(0)) THEN
PDLX(P)=TTOP(0)
FI
FI
POSPDL(P,PDLX(P),PDLY(P))
RETURN

PROC REVERSEWPK()
; Reverse horizontal puck direction
VOLUME=14
PUCKV=-1
IF PUCKV<(100-BOUNCE) THEN
PUCKV=0
ELSE
PUCKV=-(90-BOUNCE)
FI
RETURN

PROC REVERSEVPK()
; Reverse vertical puck direction
VOLUME=14
PUCKX=-1
IF PUCKX<(90-BOUNCE) THEN
PUCKX=0
ELSE
PUCKX=-(90-BOUNCE)
FI
RETURN

ANALOG COMPUTING  JANUARY 1986 / PAGE 65
PROC MOVEPUCK
BYTE PADDLE,XDIFF,YDIFF,STIK,ABOVE
CARD ARRAY
XVELC(0)=(-100 100 100 80 80 40 0
40 80 100 100 400)

; Move the puck
ERASEPUCK(PUCK)

; check for paddle collisions
PADDLE=0
IF PUCKY/100>70 THEN
PADDLE=1
FI

STIK=05TICK(PADDLE)
IF P2P1 THEN
IF HITFLAG=0 THEN
VOLUME=14

; new x velocity & direction
XDIFF=PUCKX/100+3*PDLX(PADDLE)
PUCKX=XVELC(XDIFF)
PUCKY=0

; check boundaries
IF XDIFF<5 THEN
PUCKX=5
FI

; new y velocity & direction
YDIFF=PUCKY/100+PDLY(PADDLE)
ABOVE=0

; check boundaries
IF PADDLE THEN
IF PUCKY/100*(PDLY(1)) THEN
ABOVE=1
FI
ELSE IF PUCKY/100+8*(PDLY(0)) THEN
ABOVE=1
FI
FI

; paddle not moving
IF (STIK&3)==3 THEN
PUCKY=-111

; puck not moving
ELSEIF PUCKY>0 THEN
PUCKY=200
PUCKX=0
IF (STIK&3)==1 THEN
PUCKY=-111
FI

; puck and paddle equal y coord
ELSEIF PUCKY/100+3=PDLY(PADDLE) THEN
; do nothing
ELSE IF PUCKY==1 AND (STIK&3)==1 THEN
IF ABOVE THEN
PUCKY=-200
IF PUCKX>200 THEN
PUCKY==111
FI
ELSE IF PUCKY==200
FI
ELSEIF PUCKY<0 AND (STIK&3)==2 THEN
IF ABOVE THEN
PUCKY=-200
IF PUCKX<200 THEN
PUCKY==111
FI
ELSEIF PUCKY==200
FI
ELSEIF PUCKY>0 AND

; check boundaries
IF PUCKX>24000 THEN
IF ABOVE THEN
PUCKX==111
PUCKY==1
FI
ELSEIF PUCKX<0 THEN
REVERSEPX()
PUCKX=0
ELSEIF PUCKY>BOTM100 THEN
REVERSEPY()
PUCKY=BOTM100
FI

; handle friction
IF PUCKX THEN
PUCKX=-FRICTION
FI
IF PUCKY THEN
PUCKY=-FRICTION
FI

; fading collision sound
IF VOLUME THEN
VOLUME=-2
SOUND(0,10,0,VOLUME)
SOUND(1,10,0,VOLUME)
ELSE
SOUND(0,0,0,0)
SOUND(1,0,0,0)
FI
POSSPUCK(PUCKX, PUCKY)
RETURN

PROC GOAL(PLAYER)
BYTE I
CARD J

; Inc score, check for a winner
SOUND(1)
ERASEPUCK(PLAYER)
VOLUME=0
SERVER=1
SERVER+PLAYER
SCORE(PLAYER)==+1
IF SCORE(PLAYER)==MIN THEN
GAMEOVER:=1
FI
; Flash score
FOR I:=0 TO 5 DO
SETBLOCK($2422+2*MPLAYER,2,0)
FOR J:=0 TO 5000 DO OD
SCREEN(24+2*MPLAYER):=
$16+SCORE(MPLAYER)/10
SCREEN(24+2*MPLAYER):=
SOUND(0,20,10,0)
FOR J:=0 TO 5000 DO OD
SOUND(0,0,0,0)
OD
; Cheering
IF GAMEOVER=0 THEN
FOR I:=0 TO 30 DO
FOR J:=0 TO 1000 DO OD
SOUND(0,10,0,1 RSH 1)
OD
FOR J:=0 TO 40000 DO OD
FOR I:=0 TO 30 DO
FOR J:=0 TO 1000 DO OD
SOUND(0,10,0,15-I RSH 1)
OD
FI
SNDRS() SOUNO(3,0,0,3)
RETURN

PROC MOVEALL()
; Move paddles and puck
ATTRACT=0
; Check for goal
IF PUCK<2400 AND PUCK<3700 THEN GOAL=1
ELSE IF PUCK=80 OR PUCK=100 THEN GOAL=0
FI
IF GAMEOVER=0 THEN
MOVEPUCK() MOVEPADDLE(0) MOVEPADDLE(1)
FI
RETURN

PROC ENDSGAME()
BYTE I
CARD J,K
; Cheer profusely and end game
SNDRS()
FOR I:=0 TO 20 DO
FOR J:=0 TO 1000 DO OD
SOUND(0,16,0,1 RSH 1)
OD
FOR J:=0 TO 200 DO
FOR K:=0 TO 500 DO OD
IF RAND(100)<3 THEN
FOR I:=0 TO 15 DO
FOR J:=0 TO 1200 DO OD
SOUND(1,1544,10,16-I)
OD
FOR I:=0 TO 15 DO
FOR J:=0 TO 1200 DO OD
SOUND(1,1544,10,16-I)
OD
FI
OD
FOR I:=0 TO 30 DO
FOR J:=0 TO 10000 DO OD
SOUND(0,10,0,15-I RSH 1)
OD
FOR J:=0 TO 40000 DO OD
RETURN

PROC MAIN()
LARG=0
CLOSE(1)
OPEN(1,":\\",4,0)
DO
TITLE()
WHILE CONSOLED DO OPTIONS() PLAY() OD
UNTIL KEY=27
OD
CLOSE(1)
GRAPHICS(0) GRACE=0
RETURN


CIRCLE #10 ON READER SERVICE CARD