THIS FAT ISSUE is the last for the 1980 fiscal year. The next issue, November, will be the first of Volume Three. Again, we've had a number of issues over and above the guaranteed minimum of 6 per year, and actually more pages than last year's volume. The main reason to stop at this point is to allow my wife and myself the luxury of a month's vacation to celebrate our twenty-fifth anniversary. Hopefully, the extra material in this issue will tide you over until we return and have the opportunity to prepare the next issue from all the material you will have continued to send in. While we will have a house-sitter, there won't be anyone around to do anything about telephone calls. Mail will be picked up by one of the local elves, if an "A" or "ARCADIAN" appears on the envelope.

SINCE this is the end of Volume Two, it follows that subscriptions for Volume Three are now due. I'll continue the guarantee of at least 6 issues, as I have in the past. There were 5 'extras' in Volume One and 4 in Volume Two under that scheme. The rate for the next Volume will be $12.50, partly in anticipation of a postal increase rumored for November, partly to worry about the proposed 9-digit Zip Code we may have to reprogram the address list for, and partly to allow for some increase in productivity in preparing and distributing the paper. For example, it takes 2 hours of work per 100 copies, between receipt from the printer and delivery to the post office. I'd like to decrease that labor cost and mechanize the operation a little more, but that costs money. Recall that this is a part-time hobby operation, and there are other things that call upon my time.

AN INCREASE in the number of subscribers would be a more palatable way to increase revenues, so why don't you make it a personal goal to induce a Bally buddy to have his own subscription, and I'll consider a rate reduction... Each of the little projects and publications is priced to stand alone, but a peculiarly large amount of money is consumed by the little nickle and dime (quarter and half?) things like extra postage, telephone, gas to the printer, lost mail, etc., that in the aggregate become a considerable sum.

USER GROUPS are encouraged, for when a couple of owners can get together, more is accomplished than when working alone - mutual assistance, etc. We've identified a couple of these in past issues and would like more to be developed, especially as we get into more complicated subjects. To further this concept, I'll send additional copies of the ARCADIAN to the same address at a reduced rate - three or more copies to one address at $8.50 per subscription.

WHEN SENDING your subscriptions for next year, it would make things easier if you would put an "S" on the envelope and we can channel those into the right pile to get the next year's address labels made up as soon as possible.

MORSE CODE (ENHANCED & CORRECTED)

```
  10 :RETURN ;CLEAR ;NT=1
  20 BC=RND (32)b8;FC=BC+4+RND (32)b8
  30 PRINT ;PRINT "     MORSE CODE
  40 PRINT ;INPUT "   SPEED?"D
  45 PRINT ;PRINT "TYPE'WORDS''RUN' FOR CODE
  50 PRINT ;PRINT "YOUR MESSAGE:
  60 PRINT ;B=0
  70 B=B+1;N=KP;IF N=106GOTO 200
  80 @(B)=N;TV=N; IF N=31B=B-2
  90 GOTO 70
 200 CLEAR ;FC=0;BC=3;P=0;CY=40;FOR A=1TO B-1
 210 P=P+1;IF P=10P=0;CY=40
 220 N=@(A);NT=0;TV=N;PRINT " ",;NT=D
 230 IF N<44GOTO 300
 240 IF N>90GOTO 300
 250 IF N=47GOTO 300
 260 IF N>57IF N<65GOTO 300
 270 GOSUB Nb10
 275 FOR C=1TO 25bD;NEXT C
 280 NEXT A
 290 GOTO 1000
 300 TV=31;PRINT ;FOR C=1TO 25bD;NEXT C
 310 GOTO 280
```

MORSE CODE MODIFICATION by Bob Weber adds a little enhancement by adding a question at the end - press PRINT to repeat the message, press ERASE to start a new one, & press + to add to the message.

Utilize lines 440 through 1000 as written, with the two corrections added below.

```
1005 CLEAR
1010 B=B-1;NT=0;FOR C=1TO B
1020 TV=@(C);NEXT C;PRINT
1030 PRINT "END OF MESSAGE
1040 IF &(20)=16B=B+1;GOTO 200
1050 IF &(21)=16RUN
1060 IF &(20)=8B=B+1;@(B)=32;PRINT "INPUT ";GOTO 70
1070 GOTO 1040
```

Bob Weber
6594 Swartout Rd.
Algonac, MI 48001

There were two errors in the listing on p.81:
change 660 to read PRINT"Q00 Q Q Q"; RETURN
change 460 to read PRINT"Q Q00 Q Q00 Q Q00";RETURN

SALVAGE BOARD CHECKOUT described on p. 80 has a partial error in table 2. I identified chip U14 and pin 16; but it should have been chip U14 and pin 6, as was mentioned in the text.
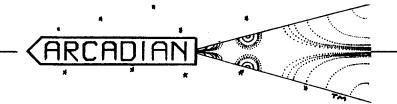
YAHTZEE ADDITION CORRECTION. The modification presented on p.82 made the dice nice and big, but I made two errors, in lines 380 and 385. Change the first command in each to have a > instead of =, so they read:
        380 IF Z>1 ...
        385 IF Z>3 ...

ATTACK MODIFICATION REBUTTAL Carl Morimoto, author of ATTACK, indicates that the modification I presented on p. 67 will cause the piece to move in two-square increments, which is not the intent. Pieces must stay within the wall constraints.

BASE CONVERSION by Ron McCoy puts it all together. Now you can have one program that will convert from any of the numerical systems into the other four, rather than have to use individual programs. If you start with decimal, the program takes the whole number in one piece and converts it. If you start with any of the other four, you have to enter each digit; as with Hex- to enter 23AC, enter 2 GO 3 GO A GO C GO and thats it.

⟨ARCADIAN⟩

```
  1 .BASE CONVERSION
  2 . RON MCCOY
  3 .   MAY 30, 1979
  4 .SZ=385
  5 :RETURN
 10 CLEAR
 20 PRINT "INPUT KNOWN BASE
 30 PRINT "    B,D,H OR O";PRINT
 40 P=KP;IF (P="B")+(P="D")+(P="H")+(P="O")GOTO 60
 50 PRINT "*INVALID-TRY AGAIN*";GOTO 240
 60 IF P="D"INPUT "INPUT #"S;GOTO 240
 70 PRINT "INPUT (W/ LEADING 0'S
 90 IF P="O"GOTO 180
100 IF P="B"GOTO 510
109 .HEX TO DECIMAL
110 A=10;B=11;C=12;D=13;E=14;F=15
120 PRINT "4 DIGIT HEX #
130 INPUT W,X,Y,Z
140 IF (W>7)+(X>15)+(Y>15)+(Z>15)PRINT "TOO BIG";GOTO 120
150 S=(4096bW)+(256bX)+(16bY)+Z
160 GOTO 220
179 .OCTAL TO DECIMAL
190 PRINT "5 DIGIT OCTAL #
190 INPUT V,W,X,Y,Z
200 IF (V>7)+(W>7)+(Y>7)+(Z>7)PRINT "TOO BIG";GOTO 180
210 S=(4096bV)+(512bW)+(64bX)+(8bY)+Z
220 PRINT ;PRINT #1,S," DECIMAL
239 .BINARY TO DECIMAL
240 Q=Sc2
250 O=RM
260 FOR R=311TO 324
270 Q=Qc2
280 GOSUB R
290 NEXT R
300 GOTO 330
311 N=RM
312 M=RM
313 L=RM
314 K=RM
315 J=RM
316 I=RM
317 H=RM
318 G=RM
319 F=RM
320 E=RM
321 D=RM
322 C=RM
323 B=RM
324 A=RM;RETURN
330 PRINT #1,A,B,C," ",D,E,F,G," ",H,I,J,K," ",L,M,N,O," BINARY
339 .BINARY TO HEX
340 Z=O+(2bN)+(4bM)+(8bL);IF Z>9Z=Z+7
350 Z=Z+48
360 Y=K+(2bJ)+(4bI)+(8bH);IF Y>9Y=Y+7
370 Y=Y+48
380 X=G+(2bF)+(4bE)+(8bD);IF X>9X=X+7
390 X=X+48
400 W=C+(2bB)+(4bA)
410 W=W+48
419 .BINARY TO OCTAL
420 V=O+(2bN)+(4bM)
430 U=L+(2bK)+(4bJ)
440 T=I+(2bH)+(4bG)
450 S=F+(2bE)+(4bD)
460 R=C+(2bB)+(4bA)
480 TV=W;TV=X;TV=Y;TV=Z;PRINT " HEX
490 PRINT #1,R,S,T,U,V," OCTAL
500 PRINT ;GOTO 20
509 PRINT "TOO BIG
510 PRINT "15 DIGIT BINARY #
520 INPUT A,B,C,D,E,F,G,H,I,J,K,L,M,N,O
530 IF (A>1)+(B>1)+(C>1)+(D>1)+(E>1)+(F>1)+(G>1)+(H>1)+(I>1)+
    (J>1)+(K>1)+(L>1)+(M>1)+(N>1)+(O>1)GOTO 509
540 S=(16384bA)+(8192bB)+(4096bC)+(2048bD)+(1024bE)+(512bF)+
    (256bG)+(128bH)+(64bI)+(32bJ)+(16bK)+(8bL)
550 S=S+(4bM)+(2bN)+O
560 GOTO 220
```

Ron McCoy
7509 W. River Rd.
Minneapolis, MN 55444

EIGHTY-NINE

BALLY WAS SPOTLIGHTED during Personal Computing 80, the fifth annual
personal computing and small business computer show, held in
Philadelphia on August 21-24. The "largest personal computing show
in 1980" devoted 2½ hours during the Visual Arts Festival held on
Sunday and sponsored by the Philadelphia Area Computer Society, to
the Bally Home Computer System. The topic, "Low Cost Pixel Art" was
presented by Zuszsanna Molnar and Frank Dietrich of Chicago Circle,
University of Illinois and Professor Lou Katz and Jinko Gotoh of
Columbia University. Discussions supported by video tape included
interactive programs for graphics, real time animation, and the new
Bally ZGRASS System. The marketing tape on the ZGRASS system ($2800
with monitor) was quite impressive: four times the video resolution
of the current Bally Arcade, 64K of memory, floating-point arithmetic
with trig functions, and other great features. Topping the show off
with a visit to the Bally Park Place Casino in nearby Atlantic City
made it a memorable weekend - (No winnings to report!)

> (above report direct from Karen Phelps Cravedi of the Tidewater
> Computer Club - I can use more ready-to-print material, thanks Karen)

PROGRAMMING HINTS listed below were extracted from an article by Steve
Kimmel in the August Creative Computing (p.64) are are submitted as
guides for the types of programs to write:
  o Don't write the program if you are going to do the job only once or
    twice a year.
  o Don't write a program if a human could do the job in less than
    5 minutes.
  o Don't write a program if you could do it better with a calendar and
    pencil, file boxes and cards, or a calculator.
  o Don't write a program if you don't need the accuracy.

TOUCHTONE modification by Rich Tietjens, now settling down at
Monterey ,CA for a few months.

```
 65 IF ((N>47)+(N<58))-((N>64)+(N<69))-(N=42)-(N=35) GOTO 70
 67 GOTO 30
287 C=119; R=188; RETURN
294 C=146; R=188; RETURN
317 C=109; R=253; RETURN
318 C=109; R=230; RETURN
319 C=109; R=208; RETURN
320 C=109; R=188; RETURN
```

BOTS program by Ron McCoy is a challenge. Modified from a program in the
June 1979 Personal Computing, it sets up a 9x19 grid on the screen.
Fifteen "walls" are randomly placed in the squares, and the 15 "bots"(*)
are also randomly placed. Then your position is similarly located.
Hand controller 1 is used to move the target (you) in any of the eight
available directions, or stand still, and the trigger makes it happen.
(The knob is used to turn a little indicator to the desired direction.)
After you make your move, all the bots start to advance upon you, one
square at a time, each. If they hit a wall, they disappear. The object
is to wipe them out, but it is a difficult job. You have to maneuver
yourself so that the bots keep hitting walls. I kept saying 'next time
I'll get them', to no avail.

```
    1 .
    2 .
    3 .
    4 .BOTS
    5 .BY RON MCCOY
    6 :RETURN
   10 CLEAR ;BC=0;FC=142;R=15
   12 PRINT "BOTS ARE *'. I'S ARE
   14 PRINT "WALLS. BOTS MOVE AT YOU,
   16 PRINT "BUT DIE IF THEY HIT A
   18 PRINT "WALL. KNOB POINTS WAY,
   20 D=600;I=650;L=700
   22 PRINT "TRIGGER MAKES MOVE.
   24 FOR A=1TO 999;NEXT A
   30 C=0;V=1;N=0;CLEAR
   40 FOR Y=-32TO 40STEP 8
   50 FOR X=-70TO 63STEP 7
   60 C=C+1;@(C)=0;BOX X,Y,6,7,1
   70 NEXT X
   80 NEXT Y
   90 FOR A=1TO R
  100 GOSUB D
  110 GOSUB I
  120 IF @(C)#0GOTO 100
  130 CX=0;CY=P;TV=73;@(C)=73
  140 NEXT A
  150 FOR A=1TO R
  160 GOSUB D
  170 GOSUB I
  180 IF @(C)#0GOTO 160
  190 CX=0;CY=P;TV=42;@(C)=1
  200 NEXT A
  210 GOSUB D
  220 GOSUB I
  230 IF @(C)#0GOTO 210
  240 FOR B=1TO 750
  250 BOX 0,P,7,7,2;LINE 0,P,4;BOX 0,P,2,2,1
  260 Q=(KN(1)+127)c28
  270 IF Q=1LINE 0,P+3,1;M=20
  280 IF Q=2LINE 0+3,P+3,1;M=21
  290 IF Q=3LINE 0+3,P,1;M=1
  300 IF Q=4LINE 0+3,P-3,1;M=-19
  310 IF Q=5LINE 0,P-3,1;M=-20
  320 IF Q=6LINE 0-3,P-3,1;M=-21
  330 IF Q=7LINE 0-3,P,1;M=-1
  340 IF Q=8LINE 0-3,P+3,1;M=19
  350 IF (Q<1)+(Q>8)M=0
  360 IF TR(1)BOX 0,P,7,7,2;BOX 0,P,6,7,1;GOTO 400
  370 NEXT B
  380 FC=0;IF TR(1)FC=142;GOTO 240
  390 GOTO 380
  400 @(C)=0;M=M+C;C=M
  410 GOSUB L
  420 CX=0;CY=P;TV=43;@(M)=43
  430 J=0;K=P;V=0;N=N+1
  440 FOR A=1TO 200
  450 IF @(A)=N C=A;GOSUB 800
  460 NEXT A
  470 IF V=1C=M;GOSUB L;GOTO 240
  480 GOSUB 750;R=R+1
  490 PRINT "HOT DAMN! YOU WON!
  500 GOTO 530
  510 GOSUB 750;R=R-2
  520 PRINT "DRAT!
  530 PRINT "WANNA RERUN?
  540 PRINT #1,"IT'LL BE ",R," BOTS
  550 PRINT "1=YES
  560 A=KP
  570 IF A="1"GOTO 30
  580 STOP
  600 X=RND (20)b7-77
  610 Y=RND (10)b8-40
  620 O=X;P=Y;RETURN
  650 T=20b(Y+32)c8
  660 S=(X+77)c7
  670 C=S+T;RETURN
  700 P=(200-C)c20+1;P=11-P
  710 O=C-(20bP)+20
  720 P=8bP-40;O=7bO-77
  730 X=O;Y=P;RETURN
  750 CX=-70;CY=CY-16;RETURN
  800 GOSUB L
  820 IF O>J X=X-7
  830 IF O<J X=X+7
  840 IF P>K Y=Y-8
  850 IF P<K Y=Y+8
  860 G=X;H=Y
  870 GOSUB I
  880 IF @(C)=43CX=G;CY=H;TV=79;GOTO 510
  890 IF @(C)=73BOX 0,P,6,7,1;@(A)=0;RETURN
  900 V=1;@(C)=N+1;CX=G;CY=H;TV=42;
      BOX 0,P,6,7,1;RETURN
```

```
  1 .COUNT THE DOTS
  2 .(C) 1980 L. & M. PORTER
  5 NT=0
  7 H=0;I=0;E=0;O=0
 10 G=0;L=0
 20 R=0;M=0;Q=0;W=0
 30 P=0;B=0;C=0;N=0
 40 CLEAR ;PRINT "INPUT # OF ROUNDS OF COUNTING ?";INPUT X
 44 IF X<1GOTO 40
 45 IF X>54NT=3
 46 IF X>54PRINT "SMALLER # OF ROUNDS ";PRINT "PLEASE!";NT=0;GOTO 40
 50 CLEAR ;PRINT "INPUT DIFFICULTY";PRINT ;PRINT "1-HARD  TO  500-EASY";INPUT D
 60 IF D>500GOTO 50
 70 IF D<1GOTO 50
 80 CLEAR ;BC=249;FC=7
 90 FOR A=1TO RND (17)
100 BOX RND (50)-25,RND (44)-22,1,1,1
110 NEXT A
120 FOR Z=1TO D
130 IF TR(1)=1GOTO 150
140 NEXT Z
150 CLEAR ;PRINT Z
155 IF Z=1NT=5;PRINT "DON'T CHEAT BY HOLDING ";PRINT "THE TRIGGER IN";PRINT "NO
W, GET READY    GO!";NT=0;GOTO 80
160 PRINT "YOUR COUNT?";INPUT K
170 Q=Q+Z
175 IF K=A-1 I=I+Z
177 IF K#A-1 W=W+Z
190 IF K=(A-1)PRINT "RIGHT!";IF K=(A-1) B=B+1
202 IF K=A-1 IF A-1>=9 P=ABS(8-(A-1))
205 IF K=A-1 IF A-1<9 P=0
207 IF A-1>=9 E=E+1
208 IF K=A-1 H=H+P
210 IF K=(A-1)G=G+(A-1)
230 IF K#(A-1)BC=93;PRINT "SORRY PAL";IF K#(A-1) C=C+1
240 PRINT "DOTS ",#1,A-1
250 PRINT "RIGHT ",#1,B
260 PRINT "WRONG ",#1,C
270 FOR J=1TO 500
280 NEXT J
285 BC=249
290 N=N+1
300 M=M+(A-1)
310 R=R+K
320 IF N=XPRINT "GAME OVER
325 O=600-D
330 L=(600-D)bX
340 V=LcQ
350 F=(Eb(Vc10)+E+B+H+G+(((ObB)-I)c10))c10
```
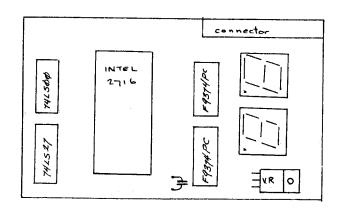
```
360 IF N=XPRINT "CORRECT COUNT=",#1,G
390 IF N=XPRINT "ADJSTD DFCLTY FCTR=",#1,V,#1,".",(RMb10)cQ
400 IF N=XPRINT "TOTAL/INPUT ",#1,M,"/",#1,R
420 IF N=XPRINT "TIME/USED ",#1,(D+1)bX,#1,"/",#1,Q
425 IF N=XIF B+G+E+H<25GOTO 465
430 IF N=XPRINT "REACT TIME/CRRCT RND=",#1,IcB
440 IF N=XPRINT "AVG REACTION TIME/RND=",#1,(W+I)cX
450 IF N=XPRINT "ALOTD/USED ",#1,((D+1)bX)cM,#1,".",(RMb10)cM,"/",#1,QcM,#1,"."
,(RMb10)cM
451 IF N=XPRINT "USED/CORRECT DOTS=",#1,QcG,#1,".",(RMb10)cG
452 IF N=XPRINT "FOR HARD RNDS=",#1,Qc(H+(9bE)),#1,".",(RMb10)c(H+(9bE))
460 IF N=XPRINT "# OF HARD RDS=",#1,E," ABS=",#1,EbB+H+G+(((ObB)-I)c10)
465 IF N=X IF X<10PRINT "10 RDS + FOR PRAC. SCORE; TRY AGAIN";GOTO 480
470 IF N=XPRINT "PRAC. SCORE=",#1,F
472 IF F>T T=F
473 IF N=XPRINT "HI SCORE=",#1,T
480 IF N=XINPUT "SAME=1  NEW=2"U
490 IF N=X IF U=1 H=0;I=0;E=0;O=0;G=0;L=0;R=0;M=0;Q=0;W=0;P=0;B=0;C=0;N=0;D=D;X
=X;GOTO 60
500 IF N=XIF U=2GOTO 5
510 GOTO 60
>
```

COUNT THE DOTS program puts up a random number of dots on the screen (max 17) for a short period of time, and as soon as you've counted them, you pull the trigger to stop the clock, and enter the amount in the key-pad. The computer will keep track of the score. "DIFFICULTY" sets the timing. By changing the X and Y valu ., you change the size of the playing field. On the surface, this seems to be just a simple game, but if we look a little deeper, we can see an application in the field of ophthalmology - checking field of view of one's eye, blind spot size, etc., and other ideas having to do with perception can be developed.

Les Porter, Box 61 Claude, TX 79019

BALCHECK For those of you technical types that obtained the material offerred in the last issue, here is a parts layout of the pc board as was produced by Bally:

```
   5 .CONNECT FOUR
   6 .BY BOB WISEMAN
  10 GOSUB 900
  20 A=1;GOSUB 200;GOSUB 700
  30 IF EGOTO 100
  40 A=2;IF B=1GOSUB 500
  50 IF B=2GOSUB 200
  60 GOSUB 700;IF EGOTO 100
  70 GOTO 20
 100 GOSUB 800;GOTO 10
 200 CX=-77;CY=40
 205 PRINT "PLAYER",#2,A,"-ENTER COLUMN ",
 210 K=KP;TV=K;X=10bK-520
 220 Y=20;GOSUB 975
 225 IF I#0MU=70;MU=49;GOTO 205
 250 RETURN
 500 H=30000;Q=-H;C=0;CX=-77;CY=40
 501 NT=0;&(21)=15;&(22)=-1;R=0
 503 PRINT "NOW I WILL FIND A MOVE ",
 505 FOR X=-30TO 30STEP 10;Y=-30;R=0
 510 GOSUB 975;IF I>0Y=Y+10;GOTO 510
 515 IF I<0GOTO 660
 520 D=4-ABS(XbY)c100;W=X;Z=Y;K=0
 525 FOR L=-1TO 1;FOR M=-1TO 1
 526 &(16)=X+30;&(17)=20bL+20;&(18)=Mb30+30
 530 X=W;Y=Z;IF L=0IF M=0GOTO 570
 535 F=3;K=K+1;P=0
 537 &(19)=Kb15
 540 X=X+10bL;Y=Y+10bM;GOSUB 975
 545 IF F=3F=I
 550 IF I<1GOTO 565
 560 IF I=F P=P+2bI-3;GOTO 540
 565 R=R+ABS(P);@(K)=P;@(K+8)=I
 570 NEXT M;NEXT L
 572 IF R=0GOTO 625
 575 FOR M=1TO 8;I=@(M);J=@(9-M)
 576 &(19)=Mb17
 580 L=J+I;P=@(M+8);R=@(17-M)
 585 IF (L=3)+(I=3)D=H
 590 IF (L=-3)+(I=-3)D=D+500
 595 IF I>0D=D+1;IF P=0D=D+1
 597 IF I#0D=D+1;IF P=0D=D+1
 600 IF (((IbI=4)b(J=0))+(L=2))b(P=0)b(R=0)D=D+100
 610 NEXT M
 615 IF (@(2)<0)+(@(3)<0)+(@(7)<0)+(@(8)<0)GOTO 620
 618 GOTO 625
 620 A=1;X=W;Z=Z+10;GOSUB 720;IF E=1 E=0;D=D-Hc10
 625 A=2;X=W
 630 IF D=QIF RND (3)=1 Q=-H
 635 IF D>Q C=W;Q=D
 660 NEXT X;X=C
 665 &(19)=0;&(18)=0;NT=1;&(20)=0;RETURN
```

Bob Wiseman
118 St. Andrews Dr.
Cincinnati. OH 45245

```
700 Y=20;W=X
705 Z=Y;I=A;GOSUB 950
710 Y=Y-10;GOSUB 975;IF I#0GOTO 720
715 Y=Z;GOSUB 950;Y=Y-10;GOTO 705
720 K=0;FOR L=-1TO 1
725 FOR M=-1TO 1;IF M=0IF L=0GOTO 750
730 K=K+1;X=W;Y=Z;P=0
735 X=X+10bL;Y=Y+10bM;GOSUB 975
740 IF I=A P=P+1;GOTO 735
745 @(K)=P
750 NEXT M;NEXT L
755 FOR K=1TO 4
760 IF @(K)+@(9-K)>2E=1
765 NEXT K;CX=CX-6;TV=32
770 X=W
790 RETURN
800 CX=-77;CY=40
805 PRINT "GAME OVER -- 4 IN A ROW"
810 PRINT "PUSH GO";K=KP;RETURN
900 CLEAR ;INPUT "#PLAYERS?"B
905 IF B>2GOTO 900
910 CLEAR ;E=0;NT=1;BC=47;FC=90
915 BOX 0,-5,71,61,1;I=0
920 FOR X=-30TO 30STEP 10
925 FOR Y=20TO -30STEP -10
930 GOSUB 950;NEXT Y;NEXT X;RETURN
950 BOX X,Y,9,9,2
955 IF I=1BOX X,Y,7,7,1;BOX X,Y,3,3,2
960 IF I=2BOX X,Y,7,3,1;BOX X,Y,3,7,1
965 MU=(X+Y+50)bI
970 RETURN
975 IF (ABS(X)>30)+(Y<-30)+(Y>20)I=-1;RETURN
980 I=PX(X,Y)+PX(X,Y+3);RETURN
```

```
1 .
2 .
3 .
4 .SPIRALS 2
5 .BY MATT GIWER
6 NT=0
10 CLEAR
100 A=3
200 B=RND (2)
210 C=RND (2)
800 X=B;Y=0
850 FOR D=1TO 100
900 X=X+B
905 Y=Y+C
907 IF X>75GOTO 100
908 IF Y>45GOTO 100
910 GOSUB 1050
950 NEXT D
990 GOTO 10
1050 .M
1060 LINE X,0,A
1070 LINE 0,-Y,A
1090 LINE -X,0,A
1110 LINE 0,Y,A
1200 RETURN
9000 .M
>
```

```
1 .
2 .AIRBRUSH
3 .AND
4 .SANDBLAST
5 .
6 .USE KNOB FOR SPRAY
7 .ANT FARM (C.SCHREIER)/C*
8 .
10 CLEAR
20 X=X+JX(1)
30 Y=Y+JY(1)
35 A=(128+KN(1))c8+1
36 B=A
40 BOX X+RND (A)-Ac2,Y+RND (B)-Bc2,1,1,3
50 GOTO 20
>
```

GRAPHICS ASSEMBLER is a programming aid for those of you who want to construct shapes using the box command. Instead of using paper and pencil to figure out where you want the various boxes, this program allows you to set up a box size, move it around the screen to where you want it, and freeze it there. The coordinates of the location are now in memory, using the following instructions:

  Use trigger to start system after pressing RUN/GO
  Enter X and Y coordinates of box size
  Enter type of box (1-4 per BOX command)
  Move joystick to desired location
  Press trigger to freeze box location.

  It will list the command to tape at any line number you select. It will also save a number of these commands and place them all on tape with a uniform line spacing. This will take place if you command "RECORD PROGRAM". This program was developed by Hugh Fitler

```
  1 .
  2 .
  3 .GRAPHICS ASSEMBLER
  4 .BY HUGH FITLER
  5 :RETURN
  6 NT=0
 10 N=0;X=0;Y=0
 20 IF TR(1) GOTO 100
 30 GOTO 20
100 CLEAR ;IF N>0GOSUB 700
105 PRINT "SIZE?
110 INPUT "X"A;IF A<1GOTO 110
120 INPUT "Y"B;IF B<1GOTO 120
130 INPUT "TYPE?"T;IF (T<1)-(T>4)GOTO 130
140 E=3
150 CY=40;GOSUB 600;PRINT " TO SAVE    BOX
155 PRINT "PRESS - KEY TO REJECT BOX
157 GOSUB 900
158 IF N>0GOSUB 700
160 X=X+JX(1);Y=Y+JY(1)
170 BOX X,Y,A,B,E;BOX X,Y,A,B,3
180 IF TR(1)E=T;GOSUB 800;GOTO 400
185 IF &(20)=4IF N=0GOTO 100
187 IF &(20)=4GOTO 460
190 GOTO 160
200 CLEAR ;CY=40;INPUT "1ST LINE #?"C
210 INPUT "LINE SPACING?"F
220 PRINT "TURN RECORDER ON AND";GOSUB 600
230 PRINT ;PRINT
240 IF TR(1):PRINT ;GOTO 260
250 GOTO 240
260 FOR M=0TO N-1
270 PRINT #2,C," BOX ",
300 FOR P=0TO 400STEP 100
310 PRINT #1,@(M+P),;IF P<400PRINT ",",
320 NEXT P
340 PRINT
345 C=C+F
350 NEXT M
355 :RETURN ;NT=0
360 PRINT
365 PRINT "TURN RECORDER OFF AND";GOSUB 600
370 IF TR(1)CLEAR ;GOTO 100
375 GOTO 370
400 @(N)=X
410 @(N+100)=Y
420 @(N+200)=A
430 @(N+300)=B
440 @(N+400)=T
450 N=N+1
460 GOSUB 850
470 PRINT "?";K=KP
480 IF (K<"1")-(K>"2")GOTO 460
490 GOTO (K-48)b100
600 PRINT "SQUEEZE TRIGGER",;RETURN
700 FOR M=0TO N-1
710 BOX @(M),@(M+100),@(M+200),
    @(M+300),@(M+400)
715 NEXT M
720 RETURN
800 BOX X,Y,A,B,T;RETURN
850 CY=40;PRINT "ADD A NEW BOX ....1
860 PRINT "RECORD PROGRAM....2
865 GOSUB 900
867 GOSUB 700
870 RETURN
900 BOX 0,CY,159,10,2
910 BOX -55,-10,50,60,2
950 RETURN
```

Hugh Fitler
628 W. Utica St.
Sellersburg, IN 47172

MORE ADD-ONS are in process to expand the utility of the Bally machine.
All these items will use the capabilities of the Blue Ram to one extent
or another. These add-ons will be available in the form of kits, as well
as completed equipment, and take the form of adapters, to be plugged
into the socket on the top of the Blue Ram. The kits will include the
supporting programs and documentation. The first kit will be for the
full size keyboard addition, and will be available on October 17.

KEYBOARD ADDITION: This kit at $24.95 provides the hardware and software
needed to attach a JAMECO 610 Unencoded Keyboard to the Blue Ram. The
JAMECO address is 1355 Shoreway Road, Belmont CA,94002, and the current
price is $34.95. Should you desire a wired and tested combination, the
package price is $89.95. Please order the keyboard direct from JAMECO-
it just saves the double shipping,etc., and we don't have to inventory
them. When the package is attached to the Bally via the Blue Ram, with
its 36" cable, you will have:

      o   Full 96 character ASCII codes, including control characters
      o   Special Words Keys
      o   Modem compatible functions
      o   Compatibility with the new Extended Basic
      o   Complete support program and documentation
      o   An attractive partial enclosure
      o   Maintenance of the Tape Cassette Interface

FUTURE ADDITIONS:
o   The next kit will be the home control module that will use the
common BSR "X-10" system. Be sure to purchase the command module that
contains the ultrasonic receiver since the mode of operation of the|Blue Ram
module will be to send the proper ultrasonic signal to the BSR unit,
which will then act as though you made a manual input. This would be
a great utilization for a salvage board, as you could program the Blue
Ram through your regular machine and its Tiny BASIC, then transfer the
Blue Ram to the salvage board (maintaining power on, of course), and
the salvage board would then perform the programmed operations.

o   Then we will provide the software necessary to have the Blue Ram
act as a compiler for music programs. Using Processor Technology
format, one can simply enter musical selections and the software will
convert that into the proper language that the Bally can understand.

o   After that, we will produce the modem addition, to allow two-
way telephone communication, using a commercially available modem
device. Again, the addition will provide the intelligence to make the
Bally and modem operate together. The Cassette Interface will be
operable with this addition, to allow material to be taped on and off
the telephone line.

As each of the above gets closer to production, we will report more
fully. The schedule now has the control module available in mid-November,
the compiler program in December. The Extended Basic cartridge is
essentially complete and should be available by the end of the year.

BLUE RAM PRICE CHANGE has become necessary because of unexpected costs which have been steadily mounting. Effective IMMEDIATELY, the $60 and $70 kits are discontinued. The full $130 kit and the wired $170 kit will be subject to a $10 increase each after October 15, 1980. We regret having to take this measure but it is necessary if we are to continue our broad support.

HYBRID PROGRAMS TUTORIAL. Hybrid programs are those wich contain both machine-code program segments and BASIC program segments. The BLUE RAM UTILITY, the BLUE RAM DIAGNOSTIC, and the resistance measuring program from the last issue are three examples of hybrid programs. Transfer to the machine-code segment is via a CALL nnnnn statement in BASIC where nnnnn is the location of the machine-code routine being referenced. Transfer back to BASIC is via a C9 return instruction in the machine-code program. The advantage of using machine-code program routines is that they run several orders of magnitude faster and can easily perform logical type operations such as ANDing and ORing. It is, however, very important to remember several NO-NO's in conjunction with hybrid programs written under the current Bally BASIC:
    1. Machine-code programs must not write into RAM areas occupied by the BASIC program. This means that many of the powerful built-in routines for graphics cannot be used without some care since the BASIC program is written in screen memory.
    2. The DE register pair of the Z80 is used by BASIC to keep track of where it is in the BASIC program. The machine-code program must return it to BASIC with the original value if the BASIC program is to resume beyond the CALL to the machine-code program.
    3. "Background" processes, running machine-code programs must also call 20B0 in order to provide BASIC with its background processing.

The last rules will be more meaningful as we proceed.

DUMPING AND LOADING HYBRID PROGRAMS is quite simple using the BLUE RAM UTILITY. The utility contains facilities for entering and dumping the machine-code portion. Bally BASIC is used in the normal way to dump the the BASIC portion. Here is a step-by-step procedure:
    1. Connect the BLUE RAM, insert the BASIC cartridge, and press RESET. Both BLUE RAM switches should be up.
    2. With the BLUE RAM UTILITY tape in the player and rewound, enter: :INPUT GO. This step will load the BLUE RAM UTILITY. Note that there seems to be a pause after a the first statement. Since the first statement does not have a line number it is executed immediately upon receipt. The effect of this statement is to continue to load data from the tape and place them in the BLUE RAM memory. They are not printed and therefore nothing is seen as they are
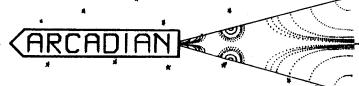
loaded. The &(192)=0 in the statement places the BLUE RAM
memory in the RAM mode so that it can be written into. The
&(64)=0 at the end of the machine-code segment load switches
the mode back to ROM. It is important to remember that just
loaded memory is left in the ROM mode and cannot be written
into further without switching it back to RAM either manually
or by program statement. The load is complete when the
screen is cleared except for "BLUE RAM UTILITY   6000" at
the top. Stop the tape at this point.
     3. Once the BLUE RAM UTILITY is loaded it can be used
to enter a new machine-code program or modify an existing one
(already in BLUE RAM memory). Note that the utility switches
the memory mode to RAM and back to ROM when changing memory
contents so it is unneccessary for you to ensure that memory
is in the RAM mode.
     4. The machine-code segment of a hybrid program should
normally be written to tape first since most changes to a
program are absorbed by the BASIC segment. To dump the
machine-code segment, enter: PRINT followed by the
beginning and ending addresses of the segment (in hexadecimal
format), followed by the letter L to indicate to the loader
that another segment (the BASIC segment) follows this one.
The utility will instruct you to start the tape RECORDING
and press GO when it has started moving and is past the
leader, if any. The utility now proceeds to write the
loader on the tape followed by the data. Since the data
is machine-code and not ASCII characters, most of the data
will show up on the screen as question marks and words with
no context. This is the data that you don't see load when
the machine-code segment is being loaded. When "END OF DUMP"
is seen on the screen, stop the tape but do not rewind it.
     5. Press RESET to clear the utility program from BASIC
memory. You are now ready to enter and dump the BASIC
segment in the normal manner using the :PRINT ;LIST
functions of BASIC.

A few additional notes concerning the utility: The machine-
code segment for the utility is at addresses 6C00 through
6CA3. The addresses for the machine-code segment of the
diagnostic are 7000 through 707E. Use these numbers in
conjunction with the utility to dump extra copies of the
utility and diagnostic. Also, by deleting the comment lines
from the BASIC segment of the utility (lines 1 through 7),
the program will run a little faster. Extra characters in
a BASIC program nearly always make it run slower.

HYBRID PROGRAM DATA TRANSFER is accomplished via the letter
variables A through Z. The BASIC program can place parameter
data into the variables prior to CALLing the machine-code
routine. Likewise, the machine-code routine can return data
to the BASIC program via the same variables. The letter
variables begin ·at address 4E6E for A and continue by two
for each succeeding letter (4E70 for B, 4E72 for C, etc.)
Hexadecimal digits have 16 possible values remember:
0123456789ABCDEF so that the difference between 4E6E and
4E70 is 2. Full word Z80 register pairs should be used to
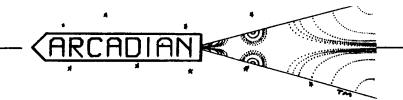access these variables since BASIC treats them as full word

integers.   Consider the following example:

```
6000 21 6E 4E 6E 26 00 22
6007 70 4E C9

10 A=28672;CALL 24576;PRINT B
```

The BASIC program sets up an input address in variable A
and calls the machine-code program (24576=6000 Hex). The
machine-code program reads a byte from the memory location
given to it in A and returns the value of that byte in B
where the BASIC program can access it to print it. For
those not familiar with machine-code programming, many
excellent texts are available in self-teaching and cook-
book style. Also, as people contribute useful machine-
code routines, their practical use will be described and
they will be added to a library of such routines. It is,
therefore, not neccessary to understand how a machine-code
routine works in order to use it in a hybrid program.
You only have to know what it does and how to call it.

BACKGROUND / FOREGROUND TUTORIAL. One of the best uses of
a machine-code routine in a hybrid program is as a back-
ground processor. The virtual effect is to split the Z80
processor into two distinct processors: a foreground
processor and a background processor. So complete is this
virtual split that the resultant two processors can talk
to each other as if they really were two separate Z80's.
Like a bi-cameral mind, each half goes about its assigned
duties totally oblivious to the other's status. The
foreground processor is responsible for the primary logic
of the overall program, gathering inputs and initiating
new processes, usually without regard for timing effects.
The background processor is triggered by an external event
and responds usually with fixed logic and parameters
supplied to it by the foreground processor. The most
common event trigger in the Bally Arcade is the screen
interrupt which occurs 60 times a second. A background
process tied to this stimulus will receive control at
this 1/60th of a second interval allowing it to perform
repetitive operations at that rate. BASIC uses such a
background process to implement the note timer NT for
the duration of the tone associated with a printed
character. The foreground processor prints the character
on the screen and then sets the value of the note timer
in a special memory location for the background processor.
After it prints the second character, it checks the special
location for a zero value. If the value is not zero it
continues to wait and check. If it is zero, it places the
new NT value in that location and continues to print the
next character. The background processor, operating at
the 60x rate decrements the special location. Therefore,
a value of NT=3 equates to a note time delay during printing
of 3/60ths of a second. The background process handles the

chore of time keeping while the foreground process is
free to continue executing the main program. The most
significant use of a background process tied to the
screen interrupt is animation. The trick of providing
smooth animation, like any magic trick, is to perform
the manipulation while no one is looking. Because of
the scanning of your TV picture, a period exists
between "frames" where the sweep is returned to the
top of the screen; a period known as vertical retrace.
During this time, no information is being written to
the screen. What you are seeing during this time is
not screen memory but the fading image of what was in
screen memory during the preceeding 60th of a second.
It is during this time, therefore, that no one is
"looking" at screen memory. By asking for the screen
interrupt to occur at the end of the scan, about three
milliseconds are available to change screen memory
before it starts being displayed again. If changes
are made during this time, they will appear as smooth
transitions as opposed to the regged-edged motion we
often see when trying to attempt high-speed animation
with the foreground BASIC program. The background
processor gives you both screen synchronization and
high speed data manipulation required for smooth,
fast animation. This is the same technique used in
the animation of the various game cartridges and now
the BLUE RAM offers you this same method. The following
tutorial describes a sample background program.

FAST ACTION GRAPHICS is demonstrated in this machine code
background routine. Enter this program using your BLUE RAM
UTILITY and watch the graphic bounce around the screen without
disturbing what is there! The speed of motion is controlled
by the #1 knob. Another special effect produced by this
routine is a "curtain of invisibility" at the top of the screen.
As the graphic moves behind the curtain, it disappears one-
half pixel at a time. The size of the curtain is also controlled
by the #1 knob. Notice that you can still enter programs, etc.
in BASIC because the graphic movement is performed in the
background mode where it does not tie up the processor.

```
6000 F3 D9 3E 60 ED 47 3E        603F FD E1 DD E1 E1 D1 C1
6007 18 D3 0D 21 6A 60 11        6046 F1 ED 7B 7E 70 C9 1F
600E 00 70 01 0D 00 ED B0        604D 00 70 56 60 08 00 98
6015 D9 FB C9 1A 60 CD B0        6054 00 50 00 00 02 08 28
601C 20 ED 73 7E 70 31 7E        605B 00 82 00 82 00 A8 2A
6023 70 F5 C5 D5 E5 DD E5        6062 80 80 80 A8 00 80 00
602A FD E5 DB 1C 32 02 70        6069 2A 20 80 00 05 00 00
6031 FF 00 07 4C 60 3F 00        6070 00 03 05 00 00 00 03
6038 70 52 60 07 4C 60 02
```

Location 6000 disables the current background process.
6001 through 6009 reassign background processing to a
routine whose beginning address is can be found in the
word at location 6018. 600A through 6014 move a motion
control table into the 7000 area so that this routine
can operate in the ROM mode if desired. 6015 & 6016
reenable the background processor. 6017 returns to
the foreground program from which it was called.
Loactions 601A through 601C envoke the original BASIC
background processor. This is necessary so that the
note timer can be decremented as required when printing
characters. Otherwise, the foreground program would
hang up waiting for the timer to clear and not be able
to continue with the foreground program. 601D through
602B save all of the Z80 registers so that their original
values can be returned to the foreground program. 602C
through 6030 read the #1 controller knob and place the
value in the motion control table as the current motion
velocity multiplier. 6031 through 603E and 604C through
6051 invoke special graphics and animation services
built into the Bally Arcade internal ROM. These will be
discussed in later issues. 603F through 604B restore
the Z80 registers to their original values and return to
the forground program in progress. 6052 through 6055
define the boundries of the graphic motion (152 wide and
40 high). 6056 through 6059 define the coordinate
reference of the graphic (0,0) and the size of the
graphic (8x8 pixels). 605A through 6069 is the image of
the graphic itself. Note that only odd bits are used.
This is very important when working with this BASIC
since even bits store the actual BASIC program statements.
If these even bits are disturbed, the BASIC program will
be altered, usually catastrophically. The remainder of
the program, 606A through 6077 define the constants for
the motion control table. These values define 5/256ths
of a pixel times the current velocity multiplier worth
of motion in both the X and Y coordinates with each
update (60 times per second). The fastest speed,
therefore, is 5 pixels x 60 updates = 300 pixels per
second. The "3's" in the table indicate that the
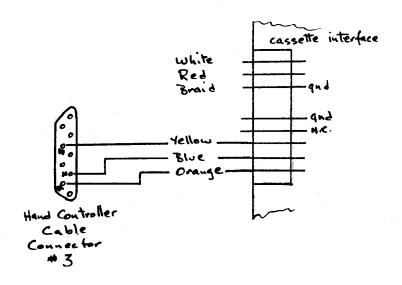graphid is to bounce off the sides instead of stopping.

To stop the background process from the foreground
process using BASIC, enter: :RETURN;GO. To start it up
again, enter: CALL 24576;GO

ADDITIONAL BASIC FEATURES are still being found. The
:INPUT and :LIST operations can be followed by an
identifier for loading or listing a particular program
from a tape that contains several programs. When dumping
these programs to tape the identifier must be placed on
the tape as follows: :PRINT ;TV=n;TV=m;LIST ... where
(m)x256+(n) is the identifier. To have BASIC automatically
find that identifier on tape, use: :INPUT (i) where i is
the same identifier. Example: :PRINT ;TV=3;TV=2;LIST
:INPUT 515 or :LIST 515.

UNLUCKY 8000 is a hex decimal value that BASIC cannot perform arithemetic on. It is equivalent in decimal value to -32768, outside BASIC's legal range. This only presents a problem when the 8000 value is required in a machine-code program as it is at locations 6066 and 6067 in the moving graphic program. Attempts to generate this number will work. It is during the read process that a HOW? error is produced. There is no known work-around for this problem except to avoid reading such a word. As was stated, you can generate it, write it to tape, and even re-read it from tape without problem. But you cannot list it or readit individually using the utility.

CASSETTE INTERFACE CABLE The schematic is shown for the cable that goes between the interface and the Bally itself, for those of you who have had problems in that area:

White
Red
Braid
gnd

gnd
NC.

Yellow
Blue
Orange

cassette interface

Hand Controller
Cable
Connector
3

HALLOWEEN GHOST is a self-running program that is topical. The author sets up a TV at a window near his front door, and the program, a "talking" skull, makes snide remarks about the people going by. It re-cycles, and uses random statements on the screen, shifting colors as well.

```
   1 .
   2 .
   3 .
   4 .HALLOWEEN GHOST
   5 .BY J.WILKINSON
   8 :RETURN
   9 FOR Z=1TO 3000;NEXT Z
  10 CLEAR ;BC=RND (256)-1
  20 FC=BC+4+RND (32)b8;GOSUB 1748
  50 X=-32;Y=18;BOX X,Y,30,2,3
 110 Y=16;BOX X,Y,34,2,3
 130 Y=14;BOX X,Y,38,2,3
 150 Y=12;BOX X,Y,42,2,3
 170 Y=10;BOX X,Y,46,2,3
 190 Y=0;BOX X,Y,50,17,3
 210 Y=-10;BOX X,Y,46,3,3
 230 Y=-13;BOX X,Y,42,3,3
 250 Y=-16;BOX X,Y,38,3,3
 270 Y=-22;BOX X,Y,34,10,3
 290 Y=-28;BOX X,Y,30,1,3
 310 Y=-29;BOX X,Y,26,1,3
 330 Y=-30;BOX X,Y,22,1,3
 350 Y=-31;BOX X,Y,18,1,3
 500 .R/SKT
 510 X=-23;Y=8;BOX X,Y,3,2,2
 520 Y=0;BOX X,Y,10,14,2
 530 Y=-8;BOX X,Y,4,2,2
 800 .L/SKT
 810 X=-41;Y=8;BOX X,Y,4,2,2
 820 Y=0;BOX X,Y,10,14,2
 830 Y=-8;BOX X,Y,3,2,2
1200 .NOSE
1220 X=-32;Y=-10;BOX X,Y,2,2,2
1240 Y=-11;BOX X,Y,4,2,2
1260 Y=-12;BOX X,Y,8,2,2
1280 Y=-13;BOX X,Y,6,2,2
1300 Y=-14;BOX X,Y,10,2,2
1320 Y=-16;BOX X,Y,6,1,2

1400 .MOUTH
1420 X=-32;Y=-24;BOX X,Y,10,2,2
1440 X=-26;Y=-22;BOX X,Y,2,2,2
1460 X=-24;Y=-20;BOX X,Y,2,2,2
1510 X=-38;Y=-22;BOX X,Y,2,2,2
1530 X=-40;Y=-20;BOX X,Y,2,2,2
1600 .R/EYE
1610 X=-26;Y=-4;A=RND (5)
1615 X=X+A;Y=Y+A;BOX X,Y,4,6,1
1620 B=RND (8)
1621 IF B=1GOTO 1640
1622 IF B=2GOTO 1740
1623 IF B=3GOTO 2000
1624 IF B=4GOTO 2010
1625 IF B=5GOTO 2020
1626 IF B=6GOTO 2030
1627 IF B=7GOTO 1740
1628 IF B=8GOTO 5
1640 FOR Z=1TO 2000;NEXT Z
1641 BOX X,Y,4,6,2;GOTO 1600
1700 .L/EYE
1710 J=-44;K=-3;C=RND (5)
1715 J=J+C;K=K+C;BOX J,K,4,6,1
1720 E=RND (8)
1721 IF E=1GOTO 1740
1722 IF E=2GOTO 1640
1723 IF E=3GOTO 3000
1724 IF E=4GOTO 3010
1725 IF E=5GOTO 3020
1726 IF E=6GOTO 3030
1727 IF E=7GOTO 3040
1728 IF E=8GOTO 5
1740 FOR Z=1TO 2000;NEXT Z
1741 BOX J,K,4,6,2;GOTO 1700
1748 CX=25;CY=20;PRINT "* * *
1760 CX=-2;CY=10;PRINT "HORRIBLE HARRY

1780 CX=0;CY=0;PRINT "THE INSULTING
1800 CX=17;CY=-10;PRINT "TV GHOST
1820 CX=6;CY=-20;PRINT "MYSTERIOUSLY
1840 CX=22;CY=-30;PRINT "APPEARS
1860 CX=-62;PRINT "--HERE EVERY HALLOWEEN--";FOR Z=1TO 1500;NEXT Z;RETURN
2000 CY=32;PRINT "  WHAT IS YOUR PROBLEM?--";GOTO 1740
2010 CY=32;PRINT "  YOU LOOK TERRIBLE!!!---";GOTO 1740
2020 CY=32;PRINT " THAT CAN'T BE -YOUR FACE-";GOTO 1740
2030 CY=32;PRINT " YIPES!!-YOU SURE ARE UGLY";GOTO 1740
3000 CY=32;PRINT "  YOUR WORMS ARE SHOWING-";GOTO 1640
3010 CY=32;PRINT "  ... WHO DUG YOU UP? ...";GOTO 1640
3020 CY=32;PRINT " HOW COME YOU HAVE 3 EYES?
3021 X=-32;Y=12;BOX X,Y,7,9,2;BOX X,Y,10,3,2;BOX X,Y,3,4,1;GOTO 5
3030 CY=32;PRINT " YOU LOOK LIKE THE -DEVIL-";GOTO 1640
3040 CY=32;PRINT " I'D HATE TO BE YOUR MUMMY";GOTO 1640
>
```

ONE HUNDRED FIVE

ONE HUNDRED SIX

ARCADIAN

Robert Fabris,vacationing
3626 Morrie Dr.
San Jose, CA 95127
Source TCD 959

HOT FLASH!!! Negotiations have been completed for Astrovision of Ohio to take over the Arcade from Bally. I expect to include a letter in the next issue that describes Bally's position with regard to the Arcades purchased from them. I understand that it will indicate their policy of maintaining warranty provisions, etc., so that the unit has not become an "orphan". I have also heard that Astrovision plans to increase the game-playing side of Arcade operation, altho they are looking at the feasibility of producing a Keyboard/Memory addition. Availability, cost, features, etc., of this are 'way in the future.