No.2 March

****
The National
Imagination Club
****

ANNOUNCEMENTS

Those of you who have tried to reach Bill Badger and have met with either no answer or just an answering machine, Protecto extends their deepest apologies. Bill has been transfered to a new position within Protecto. Those of you wishing to talk to someone about programming problems or needing technical assistance can contact Ken Whitmire at 312-695-7446. Do not try and call Bill anymore as he is unavailable.

Some of you have had questions about the Free club information you recieved with your computers. This is an inactive club in New York not The National APF Imagination Club. Only members of the National club are eligible for special price discounts as outlined in the protecto price sheets.

Questions and answers

Some of you have been wondering how to get a logical "AND" or "OR" function for IF....THEN statements. Here is one way to get them:

"AND"

The logical "AND" is usually written

```
IF J=7 AND I=6 THEN 500
```

which means both J has to be equal to 7 and I has to be equal to 6 before the program will jump to line 500.

The Imagination Machine uses a multiple IF statement to perform the same function.

It looks like this:

```
IF J=7 THEN IF I=6 THEN 500
```

"OR"

The logical "OR" is usually written

```
IF J=7 OR I=6 THEN 500
```

which means either J has to be equal to 7 or I has to be equal to 6 in order for the program to jump to line 500. The Imagination Machine can only check for numbers not letters, and this is performed like this:

```
IF (J-7)*(I-6)=0 THEN 500
```

which means that if either J is equal to 7 or I is equal to 6 the formula will equal 0 and the program will jump to 500.

*NOTE

You can not subtract a character string and get 0. Therefore this method will only work with numbers.

Some requests have been made for books showing how to use the 6800 machine language. Here are some you may want to try:

Motorola 6800 Programming Reference Manual

source: Motorola Semiconductor Products
Box 20912
Phoenix, AZ 85036

The 6800 Microprocessor: A Self Study Course with applications

by Lance Leventhal

6800 Assembly Language Programing

by Lance Leventhal

source: Osborne & Associates, Inc.
P.O. Box 2036
Berkeley, CA 94702
(415) 548-2805

How do you set random musical notes?
It would be very easy if you could just get a random number between 1 and 7, set a variable to it, and place a MUSIC command in front of it like so:

```
10 A=INT(RND(0)*7)+1
20 MUSIC A
```

The reason this doesn't work is that the music function only works with a string variable.(A$) Therefore we have to make our random number a string number between 1 and 7. (ie "5") we do this like so:

```
10 DIM A$(1)
20 J=INT(RND(0)*7)+49
30 A$=CHR$(J)
40 MUSIC A$
```

Line 10 dimensions our string variable to be at least 1 character in length.

Line 20 gets a random number between 49 and 55 and makes sure it is an integer.
Line 30 makes A$ equal to the string character whose ascii code is equal to J. If you look on page 21 of the language reference guide you will see that J is equal to a number between 1 and 7
Line 40 simply performs the music function

Try this program

```
10 DIM A$(1)
20 J=INT(RND(0)*7)+1
30 X=INT(RND(0)*31)+1
40 Y=INT(RND(0)*15)+1
50 SHAPE=15
60 COLOR=J
70 PLOT X,Y
80 K=INT(RND(0)*7)+49
90 A$=CHR$(K)
100 MUSIC A$
110 GO TO 20
```

```
10 rem just a box of checkers -
11 rem no 'kinging' allowed (yet)
15 goto 5000: rem branch to mainline
5000 rem mainline control
5002 dim cb(6,7):k$(1): rem board mtrx + xtra variables
5004 poke 8193,60: rem mode set-up
5010 poke 8194,158: rem mode2; remve 4 tst/debug
5015 rem bld shapes:bld board:put chkrs
5020 gosub 600:gosub 500: gosub 400
5025 xa=0:ya=0:x=0:y=0: rem re-init co-ordinates
5030 gosub 145: music'15': rem output cursor
5035 p=2: rem left player
5040 gosub 300: rem play game
5045 if k$="!" then 5060: rem play again?
5050 if p=2 then p=1: goto 5040
5055 goto 5035: rem left again
5060 call17046:poke 8194,30:poke 40960,3:poke 40961,224:rem crsr @ lo-lft
5065 input "want to play again? (y/n)",k$
5070 if k$="y" then 5010
5075 stop
6000 rem variables used by this pgm:
6001 rem x,y - loop controls, pointers to matrix 'cb'
6002 rem xa,ya,ra - row/col co-ordinates to shape table
6003 rem and/or screen map; row additive
6004 rem sb,sx - start/stop pointers to shape table
6005 rem sh,sl - hi/lo shape color values
6006 rem bb,by,bg,br,yy,yb,gb,rb - are shape #'s and
6007 rem indicate color of shape (color table 0)
6008 rem i.e. bb=blue/blue; gb=green/blue; br=blue/red
599:
600 rem build shapes
601 sb=512:sx=519:rem shape tbl starts at loc 512
602 gr=0:ye=85:bl=170:rd=255: rem grn,yelo,blu,red
605 xa=0:ya=0:ra=8:rem set additives
610 for x=1to8: rem 8 shapes
615 on x gosub 625,630,635,640,645,650,655,660
620 sb=sb+16:sx=sb+7:nextx: rem next block
625 sh=bl:sl=0:gosub125:bb=0:return
630 sl=ye-bl:gosub125:by=1:return
635 sl=(-bl):gosub125:bg=2:return
640 sl=rd-bl:gosub125:br=3:return
645 sh=ye:sl=0:gosub125:yy=4:return
650 sl=bl-ye:gosub125:yb=5:return
655 sh=gr:sl=bl:gosub125:gb=6:return
660 sh=rd:sl=bl-rd:gosub125:rb=7:return
499:
500 rem build board
501 sh=bb:sl=0:rem start w/ blu sqr
505 sb=0:sx=3:xa=0:ya=0:ra=32:rem set start, controls
510 for y=0to5: rem only 6 rows
515 for x=0to7: rem but 8 cols
520 gosub125:cb(y,x)=sh:rem init mtrx
525 if sh=bb then sh=yy: goto535: rem swtch color
530 sh=bb
535 xa=xa+4:nextx
540 ya=ya+64:xa=0:rem re-init 'x' additive
```

```
545 if sh=bbthen sh=yy:nexty
550 sh=bb:nexty:return
399:
400 rem put checkers on board
401 ya=0:xa=0:ra=32:rem set r/c additives
405 sh=bg:sl=4:rem strt w/ grn chkr;lo-shape is 4 away
410 for y=0to5:rem only 6 rows this board
415 for x=0to7step2
420 if y=0then435
425 on y goto 430,435,430,435,430,435
430 if x=0 then x=1:xa=xa+4
435 gosubl20:cb(y,x)=sh:rem save ckr id in mtrx
440 xa=xa+8:nextx
445 ya=ya+64:xa=0
450 if y=1 then y=3:ya=ya+128:sh=br:rem swtch chkr colr
455 nexty:return
299:
300 rem read playrs move
305 k$=key$(p):if k$=""then305:rem wait
310 if k$="!"then return:rem back to main
311 if k$="0"then return
315 if k$="?"then gosubl60:gosubl00:gosubl45:goto300:rem rmv chkr
316 if k$="5"then gosubl65:gosubl00:gosubl45:goto300:rem rpl chkr
320 k=asc(k$)-48:gosubl00:rem rmv crsr
325 on k gosub200,205,215,225,,235,240,245,255
326 rem test overflow
330 if x>7 if k=9then ya=ya+64:y=y+1
335 if x>7 if k=3then ya=ya-64:y=y-1
340 if x>7thenx=7:xa=28:music"/7/7"
345 if x<0 if k=7then ya=ya+64:y=y+1
350 if x<0 if k=1then ya=ya-64:y=y-1
355 if x<0thenx=0:xa=0:music"/1/1"
360 if y>5 if k=3 then xa=xa-4:x=x-1
365 if y>5 if k=1 then xa=xa+4:x=x+1
370 if y>5 then y=5:ya=320:music"*5*5"
375 if y<0 if k=9 then xa=xa-4:x=x-1
380 if y<0 if k=7 then xa=xa+4:x=x+1
385 if y<0 then y=0:ya=0:music"77"
390 gosubl45:goto300
199:
200 rem adjust x/y additives
201 x=x-1:xa=xa-4
205 y=y+1:ya=ya+64:return
215 x=x+1:xa=xa+4
220 y=y+1:ya=ya+64:return
225 x=x-1:xa=xa-4:return
235 x=x+1:xa=xa+4:return
240 x=x-1:xa=xa-4
245 y=y-1:ya=ya-64:return
255 x=x+1:xa=xa+4
260 y=y-1:ya=ya-64:return
100 rem most frequently called sub-rtns follo
102 rem set hi/lo shape from mtrx
105 sh=cb(y,x)
110 if sh>=by if sh<yy then sl=4:gotol20
115 sl=0
```

```
120 sb=1:sx=2:rem crsr/chkr size
125 for i =sb to sx: rem put obj on brd/scrn
130 pokei+xa+ya,sh:rem put hi shape
135 pokei+xa+ya+ra,sh+sl:rem put lo shape
140 nexti:return
145 sh=by:sl=4:rem ylo crsr
150 if cb(y,x)=yythen sh=yb:sl=(-4):rem blu crsr
155 gosubl20:return
160 cb(6,0)=cb(y,x):cb(y,x)=0:return:rem save chkr
165 cb(y,x)=cb(6,0):return:rem rplc chkr
```

checkers facility

submitted by Don Schmidt
Neptune, N. J.

instructions:

1. load program, type 'run,' Return Key.
   a) program builds board, puts checkers on board, and a yellow cursor.

2. 'left' starts.

3. digits 1-9 (except 5) control movement.
   a) move cursor to desired square, hit 'c)' key to 'mark,' and remove checker.
   b) move cursor to desired square, hit '5' to put 'marked' checker at this new location. If opponent's player was 'jumped,' position cursor on 'jumped checker' and hit '5' again to remove the checker.

4. Hit '0' (zero) to switch players.

5. Hit 'en' to end game.