FANTASMAGORICAL

◆ *Amstrad CPC Range* ● *A. R. Walker*

Perhaps the amount of space allotted to this program listing is the best indication of the quality of it.

Fantasmagorical is a platform adventure where your quest is to return one of the holy halos to the totem pole of Torag. That sounds like fairly typical adventure stuff but the platform element of the game really lifts this particular program above its contemporaries.

You play the part of Fanto Freddie, whose task it is to find the various halos littered about the realms of the castle fantasmagorical. His movements are controlled by either the cursor or a joystick and due to the high risks involved he has three lives. Despite the length of the program it is well worth the effort of typing-in all the data, but once you begin to play beware of the spiders.

```
10 REM Fantasmigorical
20 REM
30 REM ANDREW WALKER
40 :
50 KEY 159,"paper 0:pen 1:ink 0,0:ink 1,20:
mode 2:list"+CHR$(13)
60 KEY DEF 68,1,159
70 RANDOMIZE TIME
80 MEMORY &8FFF
90 DEFINT a-z:DEFREAL t
100 ENT 1,5,3,1,4,-3,1,5,3,1,5,-6,2
110 ENV 1,4,-3,2,2,-2,1
120 ENV 2,3,-2,1,2,-1,1,1,10,1
130 ENV 3,3,1,1,4,-1,2,3,1,2,5,-1,1
140 MODE 0:GOSUB 4110:GOSUB 410 ' Load
150 name$="Andrew":hs$="01000"
160 WHILE 1
170 MODE 0:sx=5:sy=0:gg=0
180 GOSUB 1020 ' Set Screen
190 RESTORE 11410:READ st$,st2$:st$=st$+nam
e$+":"+hs$+st2$:PEN 8:gg=1
200 WHILE INKEY$<>"":WEND:LOCATE 3,23:PRINT
"Fantasmagorical!":l=0:PEN 2
210 n=0:WHILE INKEY$="":n=(n MOD 15)+1
220 INK 8,n:SOUND 2,15*n,1
230 l=l+1:IF l=LEN(st$) THEN l=1
240 PEN n:LOCATE 20,25:PRINT MID$(st$,l,1);
:FOR z=1 TO 4:|SCROLL:CALL &BD19:NEXT
250 WEND:WHILE INKEY$<>"":WEND
260 GOSUB 700 ' Process Data
270 GOSUB 1020 ' Set Screen
280 t=FRE("")
290 WHILE playing
300 GOSUB 1180 ' Move You
310 GOSUB 2190 ' Move Monsters
320 WEND
330 IF win=1 THEN GOSUB 3700
340 GOSUB 3980
350 WEND
360 :
370 END
380 :
390 :
400 :
410 REM Load Data
420 :
430 RESTORE 9000:READ a$,s$:t=0:FOR l=0 TO
15:a=VAL("&"+MID$(a$,(l*2)+1,2))
440 INK l,a:t=t+a:NEXT:BORDER 0
450 IF VAL("&"+s$)<>t THEN MODE 2:PEN 1:INK
 1,20:INK 0,0:PRINT "ERROR in Colour Data:"
;9000:END
460 DIM sc$(5,4),mo(4,4,3),v$(5,4)
470 RESTORE 11100:FOR sx=0 TO 4:FOR sy=0 TO
 4:READ sc$(sx,sy),v$(sx,sy)
480 sc$(sx,sy)=UPPER$(sc$(sx,sy))
490 NEXT sy,sx:READ sc$(5,0),v$(5,0),sc$(5,
1),v$(5,1)
500 sc$(5,0)=UPPER$(sc$(5,0)):sc$(5,1)=UPPE
R$(sc$(5,1))
v$(5,1)=UPPER$(v$(5,1))
510 FOR sx=0 TO 5:FOR sy=0 TO 4
520 IF sc$(sx,sy)="" THEN GOTO 540
530 t=0:FOR l=1 TO 91:t=t+ASC(MID$(sc$(sx,s
y),l,1)):NEXT:IF t<>VAL("&"+v$(sx,sy)) THEN
 MODE 2:PEN 1:INK 1,20:INK 0,0:PRINT "Error
 in Screen Data Line:";(sx*5+sy+1)*10+11090
:END
540 NEXT sy,sx
550 RESTORE 9020:l=&A000:at=0:FOR z=1 TO 6:
READ a$,s$:t=0:FOR m=1 TO LEN(a$) STEP 2:a=
VAL("&"+MID$(a$,m,2)):POKE l,a:l=l+1:t=t+a:
NEXT:at=at+t:IF t<>VAL("&"+s$) THEN MODE 2:
PEN 1:INK 1,20:PRINT"Error in M/C Data Line
:";z*10+9010:END
560 NEXT
570 READ a$:IF at<>VAL("&"+s$) THEN MODE 2:
PEN 1:INK 1,20:PRINT"Error in checksums in
lines 9020 to 9080":END
580 RESTORE 9100:l=&9000:ln=9100:FOR z=1 TO
 25:at=0:FOR m=1 TO 6:t=0:READ a$,s$:FOR n=
1 TO LEN(a$) STEP 2:a=VAL("&"+MID$(a$,n,2))
:POKE l,a:l=l+1:t=t+a:NEXT:IF t<>VAL("&"+s$
) THEN MODE 2:PEN 1:INK 1,20:PRINT"Error in
 Sprite Data Line:";ln:END
590 at=at+t:ln=ln+10
600 NEXT m
610 READ a$:IF at<>VAL("&"+s$) THEN MODE 2:
PEN 1:INK 1,20:PRINT"Error in checksums , g
roup"(z-1)*80+9100"to"(z-1)*80+9170"of spri
tes !":END
620 ln=ln+20
630 NEXT z
640 CALL &A000
650 :
660 RETURN
670 :
680 :
690 :
700 REM Process Data
710 :
720 RESTORE 11100:FOR sx=0 TO 4:FOR sy=0 TO
 4:READ sc$(sx,sy),a$
730 sc$(sx,sy)=UPPER$(sc$(sx,sy)):NEXT sy,s
x
740 RESTORE 11380:sx=0:sy=0:FOR n=1 TO 2
750 READ a$,s$:t=0
760 FOR l=1 TO LEN(a$) STEP 4
770 k$=MID$(a$,l,4)
780 FOR m=1 TO 4
790 g=VAL("&"+MID$(k$,m,1)):t=t+g:IF g=15 T
HEN g=-1
800 mo(sx,sy,m-1)=g
810 NEXT m
820 IF mo(sx,sy,0)<>-1 THEN mo(sx,sy,0)=mo(
sx,sy,0)*6
830 IF mo(sx,sy,1)<>-1 THEN mo(sx,sy,1)=mo(
sx,sy,1)*24
840 IF mo(sx,sy,2)<>-1 THEN mo(sx,sy,2)=mo(
sx,sy,2)*6
850 IF mo(sx,sy,3)<>-1 THEN mo(sx,sy,3)=mo(
sx,sy,3)*24
860 sy=sy+1:IF sy=5 THEN sy=0:sx=sx+1
870 NEXT l
880 IF VAL("&"+s$)<>t THEN MODE 2:PEN 1:INK
 1,20:PRINT"Error in Monster data , Lines 1
1380 to 11390":END
890 NEXT n
900 RESTORE 11440:READ sx,sy,x,y
910 playing=1:score=0:energy=99:lives=4:poc
ket=-1:yloc=&9000:score$="00000":win=0
920 DEF FNoffscr=(yinc=1 AND yl>=6*24) OR (
yinc=-1 AND yl<=0) OR (xinc=-1 AND xl<=0) O
R (xinc=1 AND xl>=12*6)
930 xl=x*6:yl=y*24:os=yloc
940 MODE 0:LOCATE 1,23:PEN 4:PRINT"ENERGY":
LOCATE 2,24:PEN 2:PRINT energy;CHR$(8)"%"
950 LOCATE 13,23:PEN 4:PRINT"SCORE":LOCATE
13,24:PEN 2:PRINT score$
960 FOR c=26 TO 26+7*(lives-1) STEP 7:|SPRI
TE,&9000,6,24,175,c:NEXT:GOSUB 2330
970 :
980 RETURN
990 :
1000 :
1010 :
1020 REM Set Screen
1030 :
1040 ORIGIN 0,0,0,639,400-7*48,400:CLG
1050 FOR l=0 TO 90:a=ASC(MID$(sc$(sx,sy),l+
1,1))-65
1060 IF a<0 THEN GOTO 1090
1070 loc=&9000+a*24*6
1080 |SPRITE,loc,6,24,24*(l \ 13),6*(l MOD
13)
1090 NEXT
1100 IF gg=1 THEN |SPRITE,yloc,6,24,yl,xl
1110 IF gg=1 THEN IF mo(sx,sy,0)<>-1 THEN |
SPRITE,&9000+17*144,6,24,mo(sx,sy,1),mo(sx,
sy,0)
1120 IF gg=1 THEN IF mo(sx,sy,2)<>-1 THEN |
SPRITE,&9000+16*144,6,24,mo(sx,sy,3),mo(sx,
sy,2)
1130 :
1140 RETURN
1150 :
1160 :
1170 :
1180 REM Move You
1190 :
1200 xinc=0:yinc=0:jx=2:jy=8
1210 a=JOY(0)
1220 IF a=16 OR INKEY(18)=0 THEN a=0:yloc=&
9000:GOSUB 2730
1230 IF UPPER$(INKEY$)="A" THEN playing=0
1240 IF INKEY(47)=0 THEN WHILE INKEY$<>"":W
END:WHILE INKEY$="":WEND
1250 IF a>7 OR INKEY(1)=0 THEN a=a-8:xinc=1
:yloc=&9000+288
1260 IF a>3 OR INKEY(8)=0 THEN a=a-4:xinc=-
1:yloc=&9000+144
```

```
1270 IF a>1 OR INKEY(2)=0 THEN a=a-2:yinc=1
:IF xinc=0 THEN yloc=&9000
1280 IF a>0 OR INKEY(0)=0 THEN a=a-1:yinc=-
1:IF xinc=0 THEN yloc=&9000
1290 :
1300 IF FNoffscr THEN GOSUB 2050:RETURN
1310 :


1320 IF xinc=-1 THEN GOSUB 1430
1330 IF xinc=1 THEN GOSUB 1550
1340 IF yinc=-1 THEN GOSUB 1670
1350 IF yinc=1 THEN GOSUB 1790
1360 :
1370 IF xinc<>0 OR yinc<>0 THEN GOSUB 1920
1380 :
1390 RETURN
1400 :
1410 :
1420 :
1430 REM Go left
1440 :
1450 a1=ASC(MID$(sc$(sx,sy),((x1-jx)\6)+1+(
y1\24)*13,1))-65
1460 a2=ASC(MID$(sc$(sx,sy),((x1-jx)\6)+1+(
(y1+23)\24)*13))-65
1470 IF a1=19 OR a1=24 OR a2=19 OR a2=24 TH
EN xinc=0:GOSUB 2480:RETURN
1480 IF a1>2 AND a1<7 THEN xinc=0
1490 IF a2>2 AND a2<7 THEN xinc=0
1500 :
1510 RETURN
1520 :
1530 :
1540 :
1550 REM Go right
1560 :
1570 a1=ASC(MID$(sc$(sx,sy),((x1+jx+5)\6)+1
+(y1\24)*13))-65
1580 a2=ASC(MID$(sc$(sx,sy),((x1+jx+5)\6)+1
+((y1+23)\24)*13))-65
1590 IF a1=19 OR a1=24 OR a2=19 OR a2=24 TH
EN xinc=0:GOSUB 2480:RETURN
1600 IF a1>2 AND a1<7 THEN xinc=0
1610 IF a2>2 AND a2<7 THEN xinc=0
1620 :
1630 RETURN
1640 :
1650 :
1660 :
1670 REM Go up
1680 :
1690 a1=ASC(MID$(sc$(sx,sy),(x1\6)+1+((y1-j
y)\24)*13))-65
1700 a2=ASC(MID$(sc$(sx,sy),((x1+5)\6)+1+((
y1-jy)\24)*13))-65
1710 IF a1=19 OR a1=24 OR a2=19 OR a2=24 TH
EN yinc=0:GOSUB 2480:RETURN
1720 IF a1>2 AND a1<6 THEN yinc=0
1730 IF a2>2 AND a2<6 THEN yinc=0
1740 :
1750 RETURN
1760 :
1770 :
1780 :
1790 REM Go down
1800 :
1810 a1=ASC(MID$(sc$(sx,sy),(x1\6)+1+((y1+j
y+23)\24)*13))-65
1820 a2=ASC(MID$(sc$(sx,sy),((x1+5)\6)+1+((
y1+jy+23)\24)*13))-65
1830 IF a1=19 OR a1=24 OR a2=19 OR a2=24 TH
EN yinc=0:GOSUB 2480:RETURN


1840 IF a1>2 AND a1<7 THEN yinc=0
1850 IF a2>2 AND a2<7 THEN yinc=0
1860 IF a1=6 AND a2=6 AND pocket=7 THEN poc
ket=-1:MID$(sc$(sx,sy),(x1\6)+1+((y1\24)+1)
*13,1)="0":|SPRITE,&9000+6*144,6,24,y1+24,x
1:|SPRITE,&9000+7*144,6,24,175,71:score=sco
re+50:SOUND 1,0,-2,10,2,0,30:GOSUB 3440
1870 :
1880 RETURN
1890 :
1900 :
1910 :
1920 REM Print man
1930 :
1940 |SPRITE,os,6,24,y1,x1
1950 x1=x1+xinc*jx:y1=y1+yinc*jy
1960 |SPRITE,yloc,6,24,y1,x1
1970 os=yloc
1980 SOUND 1,0,10,5,0,0,30
1990 :
2000 :
2010 RETURN
2020 :
2030 :
2040 :
2050 REM Change screen
2060 :
2070 os=yloc
2080 IF y1=0 THEN sy=sy-1:y1=6*24:GOTO 2130
2090 IF y1=6*24 THEN sy=sy+1:y1=0:GOTO 2130
2100 IF x1=0 THEN sx=sx-1:x1=12*6:GOTO 2130
2110 IF x1=12*6 THEN sx=sx+1:x1=0:GOTO 2130
2120 :
2130 yinx=0:xinc=0:GOSUB 1010
2140 :
2150 RETURN
2160 :
2150 RETURN
2160 :
2170 :
2180 :
2190 REM Move monsters
2200 :
2210 mx1=mo(sx,sy,0):my1=mo(sx,sy,1)
2220 mx2=mo(sx,sy,2):my2=mo(sx,sy,3)
2230 IF mx1<>-1 THEN |SPRITE,&9000+17*144,6
,24,my1,mx1:mx1=mx1+2*SGN(x1-mx1):mo(sx,sy,
0)=mx1:|SPRITE,&9000+17*144,6,24,my1,mx1
2240 IF mx2<>-1 THEN |SPRITE,&9000+16*144,6
,24,my2,mx2:mx2=mx2+SGN(x1-mx2):my2=my2+SGN
(y1-my2)*2:mo(sx,sy,2)=mx2:mo(sx,sy,3)=my2:
|SPRITE,&9000+16*144,6,24,my2,mx2
2250 IF (x1+3)\6=(mx1+3)\6 AND (y1+12)\24=(
my1+12)\24 THEN energy=energy-3:SOUND 1,100
,2,15,1,1,1:LOCATE 2,24:PEN 2:PRINT energy;
CHR$(8);"% "
2260 IF (x1+3)\6=(mx2+3)\6 AND (y1+12)\24=(
my2+12)\24 THEN energy=energy-1:SOUND 1,255
,2,15,1,1,1:LOCATE 2,24:PEN 2:PRINT energy;
CHR$(8);"% "
2270 IF energy<1 THEN LOCATE 2,24:PEN 2:PRI
NT " @% ":energy=99:GOSUB 2480
2280 :
2290 RETURN
2300 :
2310 :


2320 :
2330 REM Lives
2340 :
2350 FOR c=26 TO 26+7*(lives-1) STEP 7
2360 |SPRITE,&9000,6,24,175,c
2370 NEXT
2380 lives =lives-1:IF lives<1 THEN playing
=0:GOTO 2440
2390 FOR c=26 TO 26+7*(lives-1) STEP 7
2400 |SPRITE,&9000,6,24,175,c
2410 NEXT
2420 PEN 2:energy=99:LOCATE 1,24:PRINT "  9
9% "
2430 :
2440 RETURN
2450 :
2460 :
2470 :
2480 REM Lost life
2490 :
2500 gap=90
2510 IF mo(sx,sy,0)<>-1 THEN |SPRITE,&9000+
17*144,6,24,mo(sx,sy,1),mo(sx,sy,0)
2520 IF mo(sx,sy,2)<>-1 THEN |SPRITE,&9000+
16*144,6,24,mo(sx,sy,3),mo(sx,sy,2)
2530 |SPRITE,os,6,24,y1,x1
2540 |SPRITE,&9000+10*144,6,24,y1,x1
2550 t=TIME+gap:WHILE TIME<t:WEND
2560 |SPRITE,&9000+10*144,6,24,y1,x1
2570 |SPRITE,&9000+11*144,6,24,y1,x1
2580 t=TIME+gap:WHILE TIME<t:WEND
2590 |SPRITE,&9000+11*144,6,24,y1,x1
2600 |SPRITE,&9000+12*144,6,24,y1,x1
2610 t=TIME+gap:WHILE TIME<t:WEND
2620 |SPRITE,&9000+12*144,6,24,y1,x1
2630 |SPRITE,os,6,24,y1,x1
2640 SOUND 1,0,20,7,1,30
2650 IF mo(sx,sy,0)<>-1 THEN |SPRITE,&9000+
17*144,6,24,mo(sx,sy,1),mo(sx,sy,0)
2660 IF mo(sx,sy,2)<>-1 THEN |SPRITE,&9000+
16*144,6,24,mo(sx,sy,3),mo(sx,sy,2)
2670 GOSUB 2330
2680 :
2690 RETURN
2700 :
2710 :
2720 :
2730 REM Pickup/Drop
2740 :
2750 IF ASC(MID$(sc$(sx,sy),((x1\6)-(x1 MOD
 6>3))+1+13*((y1\24)-(y1 MOD 24>12))))-65>1
9 AND ASC(MID$(sc$(sx,sy),((x1\6)-(x1 MOD 6
>3))+1+13*((y1\24)-(y1 MOD 24>12))))-65<24
THEN GOSUB 3530:RETURN
2760 IF pocket=18 AND ASC(MID$(sc$(sx,sy),(
(x1\6)-(x1 MOD 6>3))+1+13*((y1\24)-(y1 MOD
24>12))))-65=15 THEN playing=0:score=score+
1000:win=1:RETURN
2770 IF ASC(MID$(sc$(sx,sy),((x1\6)-(x1 MOD
 6>3))+1+13*((y1\24)-(y1 MOD 24>12))))-65=1
3 THEN SOUND 1,100,50,12,3,0:energy=99:LOCA
TE 2,24:PEN 2:PRINT energy;CHR$(8);"% ";:RE
TURN
2780 IF pocket=14 OR pocket=18 THEN GOSUB 2


970:IF f1=0 THEN RETURN
2790 ch=ASC(MID$(sc$(sx,sy),((x1\6)-(x1 MOD
 6>3))+1+13*((y1\24)-(y1 MOD 24>12))))-65
2800 IF ch=6 OR ch=8 OR ch=9 OR ch=13 OR ch
=15 OR (ch>18 AND ch<25) THEN RETURN
2810 swp=pocket
2820 IF pocket<>-1 THEN |SPRITE,&9000+pocke
t*144,6,24,175,71
2830 IF pocket<>-1 THEN |SPRITE,&9000+pocke
t*144,6,24,((y1\24)-(y1 MOD 24>12))*24,((x1
\6)-(x1 MOD 6>3))*6
2840 pocket=ch
2850 IF pocket<0 THEN pocket=-1
2860 IF pocket<>-1 THEN |SPRITE,&9000+pocke
t*144,6,24,175,71
2870 IF pocket<>-1 THEN |SPRITE,&9000+pocke
t*144,6,24,((y1\24)-(y1 MOD 24>12))*24,((x1
\6)-(x1 MOD 6>3))*6
2880 IF swp<>-1 THEN MID$(sc$(sx,sy),((x1\6
)-(x1 MOD 6>3))+1+13*((y1\24)-(y1 MOD 24>12
)))=CHR$(65+swp)
2890 IF swp=-1 THEN MID$(sc$(sx,sy),((x1\6)
```

```
-(x1 MOD 6>3))+1+13*((y1\24)-(y1 MOD 24>12)
),1)="0"
2900 swp=0
2910 SOUND 1,100,30,7,0,1
2920 :
2930 RETURN
2940 :
2950 :
2960 :
2970 REM Shoot ?
2980 :
2990 f1=0
3000 IF pocket=10 AND (mo(sx,sy,2)<>-1 OR m
o(sx,sy,3)<>-1) THEN SOUND 1,0,20,7,1,0,30:
GOTO 3120
3010 IF pocket=14 AND (mo(sx,sy,0)<>-1 OR m
o(sx,sy,1)<>-1) THEN SOUND 1,0,20,7,1,0,30:
GOTO 3030
3020 f1=1:RETURN
3030 |SPRITE,&9000+14*144,6,24,175,71:pocke
t=-1
3040 mx1=mo(sx,sy,0):my1=mo(sx,sy,1):sy1=y1
:sx1=x1
3050 |SPRITE,&9000+14*144,6,24,sy1,sx1
3060 WHILE sx1<>mx1 OR sy1<>my1
3070 |SPRITE,&9000+144*14,6,24,sy1,sx1
3080 sx1=sx1+SGN(mx1-sx1):sy1=sy1+SGN(my1-s
y1)
3090 |SPRITE,&9000+144*14,6,24,sy1,sx1:FOR
gap=1 TO 30:NEXT
3100 WEND:|SPRITE,&9000+14*144,6,24,sy1,sx1
:score=score+200:GOSUB 3440:mo(sx,sy,0)=-1:
mo(sx,sy,1)=-1:mf=17:GOSUB 3300:RETURN
3110 :
3120 |SPRITE,&9000+18*144,6,24,175,71:pocke
t=-1
3130 mx1=mo(sx,sy,2):my1=mo(sx,sy,3):sy1=y1
:sx1=x1
3140 |SPRITE,&9000+18*144,6,24,sy1,sx1
3150 WHILE sx1<>mx1 OR sy1<>my1
3160 |SPRITE,&9000+144*18,6,24,sy1,sx1
3170 sx1=sx1+SGN(mx1-sx1):sy1=sy1+SGN(my1-s
y1)
3180 |SPRITE,&9000+144*18,6,24,sy1,sx1:FOR
gap=1 TO 30:NEXT
3190 WEND:|SPRITE,&9000+18*144,6,24,sy1,sx1
:score=score+100:GOSUB 3440:mo(sx,sy,2)=-1:
mo(sx,sy,3)=-1:mf=16:GOSUB 3300
3200 x2=RND*3:x3=RND*12:y2=RND*3:y3=RND*6
3210 WHILE x2=sx OR y2=sy OR  MID$(sc$(x2,y
2),x3+1+y3*13,1)<>"0"
3220 x2=RND*3:x3=RND*12:y2=RND*3:y3=RND*6
3230 WEND
3240 MID$(sc$(x2,y2),x3+1+y3*13,1)=CHR$(10+
65)
3250 :
3260 RETURN
3270 :
3280 :
3290 :
3300 REM Sink into floor
3310 :
3320 |SPRITE,&9000+mf*144,6,24,my1,mx1
3330 FOR I=0 TO 22
3340 |SPRITE,&9000+mf*144,6,23-I,my1+I,mx1
3350 CALL &BD19:CALL &BD19
3360 SOUND 1,10*I*(mf-15),1
3370 |SPRITE,&9000+mf*144,6,23-I,my1+I,mx1
3380 NEXT
3390 :
3400 RETURN
3410 :
3420 :
3430 :
3440 REM Score
3450 :
3460 LOCATE 13,24:PEN 2:MID$(score$,7-LEN(S
TR$(score)))=MID$(STR$(score),2)
3470 PRINT score$;
3480 :
3490 RETURN
3500 :
3510 :
3520 :
3530 REM Tardis
3540 :
3560 m=32*(x1>40)-32*(x1<40):xinc=-SGN(m)
3570 |SPRITE,os,6,24,y1,x1
3580 FOR I=0 TO 23
3581 |SPRITE,os,6,24-I,y1+I,x1:|SPRITE,os+6
*(23-I),6,I+1,y1,x1+m
3602 FOR d=1 TO 40:NEXT
3603 SOUND 1,200-I*5,3,10:SOUND 3,200+I*5,3
,10
3604 |SPRITE,os,6,24-I,y1+I,x1:|SPRITE,os+6
*(23-I),6,I+1,y1,x1+m
3610 NEXT:x1=x1+m
3620 |SPRITE,os,6,24,y1,x1
3630 energy=energy-35:IF energy<1 THEN p=99
+energy:GOSUB 2480:energy=p
3640 LOCATE 2,24:PEN 2:PRINT energy;CHR$(8)
;"% "
3650 :
3660 RETURN
3670 :
3680 :
3690 :
3700 REM win
3710 :
3720 sx=5:sy=1:GOSUB 3440:gg=0:GOSUB 1020
3730 gg=1
3740 PEN 4:LOCATE 3,7:PRINT "Congratulation
s!":p=energy
3750 WHILE lives>0:FOR I=p TO 0 STEP -1:ene
rgy=I
3760 SOUND 1,255,2,15,1,1,1:LOCATE 2,24:PEN
 2:PRINT energy;CHR$(8);"% "
3770 score=score+10:GOSUB 3440:NEXT:GOSUB 2
330:energy=99:WEND
3780 t=TIME:WHILE t+300>TIME:WEND:BORDER 9
3790 MODE 0:p=4
3800 FOR I=0 TO 640 STEP 8
3810 MOVE 1,0:DRAW 1,p-2,9:PLOT 1,p,2:MOVE
1+4,0:DRAW 1+4,p-2,9:PLOT 1+4,p,2
3820 p=p+SGN((RND*6)-3):p=p-(p<1):p=p+(p>8)
3830 NEXT
3840 |SPRITE,&9000+288,6,24,191-24,2
3850 PEN 3:LOCATE 4,5:PRINT"Press Any Key."
3860 PEN 6:LOCATE 4,7:PRINT"You Are Free !"
3870 WHILE INKEY$<>"":WEND:I=632
3880 WHILE INKEY$=""
3890 |SCROLL
3900 MOVE 1,0:DRAW 1,p-2,9:PLOT 1,p,2:MOVE
1+4,0:DRAW 1+4,p-2,9:PLOT 1+4,p,2
3910 p=p+SGN((RND*6)-3):p=p-(p<1):p=p+(p>8)
3920 WEND:BORDER 0
3930 :
3940 RETURN
3950 :
3960 :
3970 :
3980 REM high score
3990 :
4000 IF score<=VAL(hs$) THEN GOTO 4070
4010 hs$=score$:FOR I=1 TO 40:PRINT:NEXT
4020 PEN 2:PRINT "You have the Highest scor
e":PRINT
4030 PEN 7:PRINT "Enter your name"
4040 PEN 4:INPUT ">",name$:IF name$="" OR L
EN(name$)>10 THEN GOTO 4040
4050 IF LEFT$(UPPER$(NAME$),6)="ANDREW" THE
N name$="Mr. Skill!"
4060 :
4070 RETURN
4080 :
4090 :
4100 :
4110 REM Instructions
4120 :
4130 FOR I=1 TO 15:INK 1,1:NEXT
4140 a$="FANTASMAGORICAL!":p=1:FOR I=1 TO L
EN(a$):LOCATE 1+2,2:PEN p
4150 p=p+1:IF p=8 THEN p=p+1
4160 IF p=16 THEN p=1
4170 PRINT MID$(a$,I,1);:NEXT
4180 LOCATE 1,1:PRINT STRING$(20,"="):PRINT
 STRING$(20,"=")
4190 PEN 15:PRINT:PRINT SPC(8)"KEYS":PEN 1:
PRINT SPC(7)"******"
4200 PEN 2:PRINT:PRINT SPC(2)"USE:Cursor Ke
ys"
4210 PEN 3:PRINT:PRINT SPC(9)"OR"
4220 PEN 4:PRINT:PRINT SPC(1)"Joystick  to
move"
4230 PEN 5:PRINT:PRINT SPC(6)"A  N  D"
4240 PEN 6:PRINT:PRINT"ENTER or FIRE BUTTON
";
4250 PEN 7:PRINT:PRINT SPC(1)"To Pickup / D
rop /"
4260 PEN 10:PRINT:PRINT SPC(3)"Use An Objec
t"
4270 PEN 11:PRINT:PRINT:PRINT SPC(4)"PLEASE
 WAIT"
4280 :
4290 RETURN
4300 :
4310 :
4320 :
9000 DATA 000119030b11141a000907080d040215,
a7
9010 :
9020 DATA 010aa0211fa0c3d1bcc912a0c323a0c36
ca05350524954c55343524f4ccc,d51
9030 DATA 001fa00aa02100c0dd460078fe0028042
33d20fcdd460278fe0028d0100,95c
9040 DATA 0009300401 50c0093d20f3dd4e04dd460
6dd5609dd5e00c5e51aaa772313,aa5
9050 DATA 10f9e1010008093004015 0c009c10d20e
8c92180c716011e01cd19bd237e,9cb
9060 DATA 2b7723147afe5020f53e007701b107093
0040150c00916011c7bfe0920e0,930
9070 DATA c9,c9
9080 DATA 3516
9090 :
9100 DATA 00003c3c000000146894280003cc0c03
c000068e8d494000,660
9110 DATA 0068c0c09400003cc0c03c00001468942
80000103c3c2000,654
9120 DATA 00b0303070005 0b0303070a0503030303
0a050b0303070a0,810
9130 DATA 15f0e0d0f02a007ae0d0b50000353a3f3
a000035303a3500,86e
9140 DATA 003a353035000030353035000030 3a303
a00003530303a00,341
9150 DATA 003a3035300010306090302030 30c0c03
0301020408010 20,5af
9160 DATA 261e
9170 :
9180 DATA 0000143c000000003c3c280000143c3c3
c000014943c3c00,2d8
9190 DATA 0040c03c3c000040c03c3c000040943c3
c0000143c3c2800,4f0
9200 DATA 00003c3c0000000030200000001030300
0000030f03a0000,292
9210 DATA d0f0f03a0000d0f0b5302a003f3f3a302
a00001030302a00,765
9220 DATA 00103035 2a00003030353f00003030353
f00103030353f00,32b
```

```
9230 DATA 103030356a2a303030035c0c01030206ac
000002000000c000,668
9240 DATA 1b52
9250 :
9260 DATA 00003c28000000143c3c0000003c3c3c2
800003c3c682800,2d4
9270 DATA 003c3cc00000003c3cc08000003c3c688
00000143c3c2800,584
9280 DATA 00003c3c000000001030000000030302
000000035f03000,28d
9290 DATA 000035f0f0e00015307af0e0001530353
f3f001530302000,711
9300 DATA 00153a302000003f3a303000003f3a303
000003f3a303020,34a
9310 DATA 15953a303020c0c03a30303040c095103
02000c000001000,673
9320 DATA 1bb3
9330 :
9340 DATA 0c080c0c0c0c0833043333330433043333
3043304333333304,2bc
9350 DATA 3304333333043304333333043304333333
3040000000000004,200
9360 DATA 040c080c0c0c0019330433220819330433
2081933043322081,237
9370 DATA 1933043322081933043322081933043322
0819330400000,257
9380 DATA 1933000c0c0819330433330419330433330
4193304333304,29a
9390 DATA 1933043333041933043333041933043333
3040800040000004,23e
9400 DATA ea2
9410 :
9420 DATA 0c080c0c080c3304333304333304333304
33333043333043,2bc
9430 DATA 3304333304333304333304333304333304
43300000000000,27c
9440 DATA 040c0c0c0c0c19333333332219333333332
2219333333332322,355
9450 DATA 1933333333221933333333221933333333
22080000000000,31d
9460 DATA 040c080c0c0c19330433322219330433322
19330433322,2c4
9470 DATA 1933043333221933043333221933043333
22080004000000,294
9480 DATA 1102
9490 :
9500 DATA 040c080c0c041933043322219193304332
219193304332219,26e
9510 DATA 1933043322191933043322191933043322
19080000000000,242
9520 DATA 000c0c0c040c0433332219220433332219
22204333322192 2,289
9530 DATA 0433332219220433332219220433332219
22000000001922,290
9540 DATA 040c040c192219221922192219221922192
19221922192219221922192219221922192219221922
192219221922192 2,26e
9550 DATA 192219221922192219221922192219221
922080008000800,22b
9560 DATA e62
9570 :
9580 DATA 1030303030200000c0c0400000000040
40000000040400,144
9590 DATA 0004040008000004040800000000c0c0
00000000c0c0000,60
9600 DATA 00000c0c0000000004080000cccc0408c
ccc3366bc4c9933,59d
9610 DATA 3333084433333336688449933cccc0000c
ccc0020000001000,729
9620 DATA 0020000010000020000010001000000000
020100000000020,c0
9630 DATA 1000000000202000000001020000000000
0102000000000 10,c0

9640 DATA fea
9650 :
9660 DATA 000033cc0000003333660000113333330
000220033333333,363
9670 DATA 0000993333220000993333220000cc333
3000000cc333300,4a6
9680 DATA 0044cc3322000044883322000044883 30
000004488330000,484
9690 DATA 00cc0022000000cc0022000000cc00000
00000cc00000000,374
9700 DATA 4488000000004488000000004488000000
0044880000000000,330
9710 DATA cc0000000000cc0000000000cc0000000
000cc0000000000,330
9720 DATA 1661
9730 :
9740 DATA fcfcfcfcfca8fca854545400fcfca8a8a
800fc545454000000,e1c
9750 DATA fc00a8a80000fca8a8a80000fc5454000
000a8fc54000000,8dc
9760 DATA a8a8a8000000fca8a8000000a8fc00000
000fc540000000000,738
9770 DATA a8fc00000000fc540000000000a8a800000
000fca800000000,5e8
9780 DATA a8a800000000a8a800000000fc0000000
000fc000000000000,498
9790 DATA fc0000000000a800000000000a800000000
000a800000000000,2f4
9800 DATA 2ba4
9810 :
9820 DATA 54fcfcfcfcfc00a8a8a854fc00545454f
cfc0000a8a8a8fc,e70
9830 DATA 0000545400fc0000545454fc000000a8a
8fc000000a8fc54,7e0
9840 DATA 000000545454000000054 54fc00000000f
c540000000000a8fc,594
9850 DATA 00000000fc540000000000a8fc000000005
4540000000054fc,4ec
9860 DATA 0000000054540000000054540000000000
0fc000000000fc,348
9870 DATA 0000000000fc000000000054000000000
054000000000054,1f8
9880 DATA 2610
9890 :
9900 DATA 8000000000400000000000000000000000
000000000000000,c0
9910 DATA 0000000000000000000000000000000000f0
0000005050f0f00,37
9920 DATA 000f0a0f0f00000f0f050a00000f0f0f0
000000f0f0a0f00,c8
9930 DATA 000505f0f0000000f0f0f000050f050
f000050f0a0a00,a5
9940 DATA 00000f000000000000000000000000000
000000000000000,f
9950 DATA 0000000000000000000000000000000000
0008000000000040,c0
9960 DATA 333
9970 :
9980 DATA 8000050a004000000f0f000000000050f0
a000000050f0a00,129
9990 DATA 000f000f0f00000f000f0f00000f0a0f0
f00050f0a0f0a00,c8
10000 DATA 050f0a0f000a050f0f0050a050f0f00
0f0a0f0f0a000f0f,eb


10010 DATA 0f0f0a050f0f050f00050f0a000a0505
0f0a00000f0a0f0a,dc
10020 DATA 0005 0f0a0f0a000f0f0a0f00000f0f0a
0500000f0f0f0500,cd
10030 DATA 0005 0f0f000000050f0f0a0000000f0f
0000600005 0a0040,13d
10040 DATA 5c2
10050 :
10060 DATA 0000050a004000000 50a00000f00050a
000f0f0a050a050f,147
10070 DATA 050a050a050a050a050a050a050a000f050a
0f00000f050a0f00,b4
10080 DATA 000f00000f000000000000000000000000
00000f0f00000f0f,5a
10090 DATA 0f0f00000f0f000000000000000000000
0000000000000f00,4b
10100 DATA 000f050a0f00000f050a0f0005 0f050a
0f0a050a050a050a,c3
10110 DATA 0f0a050a050f0f00050a000f0000050a
00000000050a0040,147
10120 DATA 4aa
10130 :
10140 DATA 0000c3c3000000410000820000041c3c3
82000000ffff0000,690
10150 DATA 0000d7ff00000000ebff00000000d7ff
00000000ffff0000,794
10160 DATA 0000ebff00000055d7ffaa000055d7ff
aa0000ebebebd700,b2c
10170 DATA 55d7ffebc3aa41ebd7ffc3aa55d7ebff
ebaac3ffd7d7ffff,130b
10180 DATA c3d7ebebffffffebffffd7ffd7c3c3ff
ffffebffebebebff,1630
10190 DATA 55c3d7ffffaa55ebffd7d7aa00d7ebeb
ff000041c3c38200,f23
10200 DATA 51ee
10210 :
10220 DATA 0000000000080008000800080040c0008
040c0000040c0000,70
10230 DATA 0000000000000000000000000000000000
00004040c040c00,64
10240 DATA 0400080000000c0000000800400000000
0008040000040 40c,64
10250 DATA 0000080400000000000c0800000040c04
0000000000040000,58
10260 DATA 0000080000000000800000000000000000
0000040c04040c00,4c
10270 DATA 0008040000000000c0800000000400
0000000040000000,40
10280 DATA 21c
10290 :
10300 DATA 0020201010000010102020000000ffff
000000 55ffffaa00,5bb
10310 DATA 00ffd7ebff000055ffffaa00005555aa
aa000000aa550000,9ba
10320 DATA 00f0f0f0f0005 0f0f0f0f0a000f0a050
f000000050a00000,c30
10330 DATA 003f3f3f3f00153f2a153f2a0000152a
00000000152a0000,276
10340 DATA 1020303010203010303020300000 1020
0000000010200000,210
10350 DATA 0000400000000000c0c0000000c0c0c0
c000c0c0c0c0c0c0,9c0
10360 DATA 29eb
10370 :

10380 DATA 004444440000 4444cccc44004488c988
cc0000c9cccc8888,98a
10390 DATA 44ccc9cccc8844cccccccc900ccc6ccc9
cc8844cccccccc44,f9d
10400 DATA 88cccddcc8844dd55aaaaccccccc6498
cccccc6cccccc88,10f7
10410 DATA 00ccc6ccc688cccccccccccc44ccccc6
c94488c9c6cccc88,105a
10420 DATA 44c6cccc688cccc9cccc4400cccc9
888844ccc6c6cc88,f8e
```

This program to be continued next month.

FEBRUARY ISSUE