

# 8章

## 第8章

# ピリリと辛いテクノ味

もうあなたが、BS-BASICで使いこなせないものはない。これからは、いかにBASICを駆使してプログラムを作るかという問題になる。

BASICによるプログラムにはいくらかのテクニックや、お決まりの手段というのがあるから、君にこっそりその秘密を教えてあげよう。

# サンプルプログラム

# お絵描きプログラム

このプログラムは、LINE文、PAINT文、PATTERN文といったグラフィック命令を用いて風景を描き、雲と自動車を少しづつ動かしたものです。初めのうちは静止した絵を描き、慣れてきたらそれを動かしてみましょう。

30~190 PATTERN文で使う画面データを読み取っている。

200~320 LINE文、PAINT文、BOX文、CIRCLE文な

どを用いて景色と家を描きます。

330~470 エンジンの音とクラクションの音を出すサブ  
ルーチン

## 480～610 雲を動かすサブルーチン

## 700~730 車を動かすサブルーチン

## 700~730 雲のデータ

## 740~800 車のデータ

```

10 REM ** GRAPHIC フロッグ **
20 COLOR,1:PRINT CHR$(6)
30 REM ** データ 34 ***
40 READ A
50 IF A=999 THEN 80
60 A$=A$+CHR$(A)
70 GOTO 40
80 READ D
90 IF D=999 THEN 120
100 D$=D$+CHR$(D)
110 GOTO 80
120 READ E
130 IF E=999 THEN 160
140 E$=E$+CHR$(E)
150 GOTO 120
160 READ F
170 IF F=999 THEN 200
180 F$=F$+CHR$(F)
190 GOTO 160
200 REM ** カメン ツカツ ***
210 LINE[2]0,120,191,120
220 PAINT[6]0,121,2
230 BOX [1]30,83,74,120:PAINT[3]32,85,1
240 LINE [5]24,91,52,55,80,91:PAINT[3]5
2,57,5,1
250 BOX[1]34,100,42,118:PAINT[7]36,102,
1:SET[1]36,110
260 BOX[1]50,90,70,109:PAINT[5]55,95,1
270 CIRCLE[6]50,90,13,1.6,3*/2,0:PAINT[6]
53,91,1,6
280 CIRCLE[6]70,90,13,1.6,,3*/2:PAINT[6]
167,91,1,6
290 CIRCLE[2]100,11,10:PAINT[2]100,11:P
AINT[5]68,103,6,1
300 CIRCLE[1]100,4 0,150,1.9,1.18*,*6/4
310 CIRCLE[1]110,35 ,149,1.9,1.19*,*6/4
320 PAINT[7]40,136,1,2
330 REM ** オト ツタズ ***
340 X1=23:X2=100:X4=180 :Y4=105
350 SOUND0,3:SOUND@15,1023
360 GOSUB 480
370 X4=X4-1:IFX4=0 THEN X4=191
380 IFX4=74 GOSUB 400:GOTO 360
390 GOSUB 620:GOTO 360
400 REM ** オト ツタズ ***
410 MUSIC@"L123CFA2R2L433CFA2"
420 FOR I=0 TO 1000:NEXT I
430 SOUND0,3:SOUND@15,1023

```

```

440 X4=X4+1
450 GOSUB 480
460 IF X4=192 THEN MUSIC@"L123CFA2R2L43
3CFA2":FOR I=1TO1000:NEXT I:RETURN
470 GOSUB 620:GOTO 440
480 REM ** クロ ノ ハコ カス **
490 G=RND(2)*10
500 IF G<=6THENRETURN
510 IF G=7THEN 530
520 IF G>7 THEN 560
530 POSITION X1,22
540 PATTERN[7]4,A$
550 X1=X1+1
560 POSITIONX2,38
570 PATTERN[7]4,A$
580 X2=X2+1
590 IF X1>=191 THEN X1=1
600 IF X2>=191 THEN X2=1
610 RETURN
620 REM ** クロ ノ ハコ カス **
630 POSITION X4,Y4
640 PATTERN[7]4,D$
650 POSITION X4 ,Y4
660 PATTERN[5,W1]3,E$
670 POSITIONX4 ,Y4+8
680 PATTERN[2,W1]4,F$
690 RETURN
700 REM ** クロ ノ ハコ カス **
710 DATA 0,0,0,0,0,0,0,0,0,6,0,0,240,31
,6,0,248,255,31,0,252,255,255,255,2
720 DATA 252,255,255,3,252,255,255,15,2
52,255,255,63,252,255,255,63,252,255,255
,63
730 DATA 248,255,255,15,240,255,199,15,
0,252,0,15,0,248,0,0,0,0,0,0,999
740 REM ** クロ ノ ハコ カス **
750 DATA 0,0,0,0,0,192,63,0,0,32,34,0,0
,16,66,0,96,8,130,0,32,4,2,1,252,255,255
,31,252
760 DATA 255,255,63,124,255,255,62,62,2
54,127,60 ,30,252,63,56,140,0,0,1,6,0,0,
0
770 DATA 0,0,0,0,0,0,0,0,0,999
780 DATA 0,0,0,0,192,63,0,224,63,0,240,
127,0,248,255,0,252,255,999
790 DATA 128,0,0,1,192,1,128,3,224,3,19
2,7,96,3,192,6,224,3,192,7,192,1,128,3
800 DATA 0,0,0,0,999

```

\* サンプルプログラム中に出てくる〔 〕はキーボード上で( )と同じです。

# GUNDAMの戦い

このプログラムは、GUNDAM-1でガンダムだけを動かし、GUNDAM-2で小ザクと大ザクを動かし、GUNDAM-3でGUNDAM-1とGUNDAM-2を合わせてガンダムとザクが戦うようにプログラムの追加をした3つの部分からなります。また最後に全体のプログラムも載せておきます。

## GUNDAM-1の解説

プログラムを全部打ち込んだら、よく確かめて RUN **RETURN** してください。

しばらくして、画面の下の方にガンダムが表示されます。ガンダムを右に動かすには "P" 左に動かすには "O" ビームライフルを発射するにはスペース・キーを押します。

ジョイスティックを使用する場合次の所を変更します。

```
630 JOYS1 JS:XX=X
650 IF JS=1 THEN 730
655 IF JS=9 THEN 700
660 IF JS=5 THEN 670
750 IF JS=9 THEN 2500
760 IF JS=1 THEN 850
770 IF JS=5 THEN 800
900 IF JS=5 THEN 980
910 IF JS=9 THEN 960
920 IF JS=1 THEN 940
```

ジョイスティックを使用した場合、ジョイスティックの1番で右、5番で左にガンダムを移動し、9番でビームライフルを発射します。

プログラムの行番号と内容はそれぞれ次のようになります。

200～290	ガンダムの形を出力したり消したりする サブ・ルーチン
600～646	メイン・ルーチン
650～980	ガンダムを左右に動かすサブ・ルーチン
2500～2540	ビーム・ライフルのサブ・ルーチン
2900～2930	ビーム・ライフルの発射で音を出すサブ・ ルーチン
4120～4340	ガンダムのいろいろな形の画面データと その読み込み

ガンダムが左右に動きビーム・ライフルを発射できたら  
SAVE "GUNDAM-1" **RETURN** でカセットテープにセ  
ーブします。

\* サンプルプログラム中に出てくる〔 〕はキーボード上で( )と同じです。

## GUNDAM-1

```
110 PRINT CHR$(6)
120 GOTO 4120
200 POSITION XX,150: PATTERN[0]2,LL$:RET
URN
210 POSITION X,150: PATTERN[7]1,GA$:RETU
RN
220 POSITION X,150: PATTERN[7]2,LL$:RETU
RN
230 POSITION XX,150: PATTERN[0]1,GA$:RET
URN
240 POSITION X,150: PATTERN[7]2,RR$:RETU
RN
250 POSITION XX,150: PATTERN[0]2,RR$:RET
URN
260 POSITION XX,150: PATTERN[7]1,L1$:RET
URN
270 POSITION XX,150: PATTERN[0]1,L1$:RET
URN
280 POSITION XX,150: PATTERN[7]1,R1$:RET
URN
290 POSITION XX,150: PATTERN[0]1,R1$:RET
URN
600 REM ** MAIN **
610 X=90:GD=2:CT=0
625 GOSUB 210
630 USR($00D8):GET KE$:XX=X
640 ON GD GOSUB 650,750,900
645 IF CT>500 THEN END
646 GOTO 630
650 IF KE$="P" THEN 730
655 IF KE$=" " THEN 700
660 IF KE$="O" THEN 670
665 GOSUB 200:GOSUB 260:GOSUB 270:GOSUB
210:GD=2:RETURN
670 X=X-4:IF X<4 THEN X=4:GOSUB 200:GOS
UB 210:GD=2:RETURN
675 GOSUB 200:GOSUB 220:RETURN
700 GOSUB 200:GOSUB 210:GD=2:GOSUB 2500
:RETURN
730 GOSUB 200:GOSUB 210:GOSUB 230:GOSUB
240:GD=3:RETURN
750 IF KE$=" " THEN 2500
760 IF KE$="P" THEN 850
770 IF KE$="O" THEN 800
780 FOR I=0 TO 50:NEXT:RETURN
800 X=X-4:IF X<4 THEN X=4:GOSUB 230:GOS
UB 210:RETURN
810 GOSUB 230:GOSUB 260:GOSUB 270:GOSUB
220:GD=1:RETURN
850 X=X+4:IF X>174 THEN X=174:GOSUB 230
:GOSUB 210:RETURN
860 GOSUB 230:GOSUB 280:GOSUB 290:GOSUB
240:GD=3:RETURN
900 IF KE$="O" THEN 980
910 IF KE$=" " THEN 960
920 IF KE$="P" THEN 940
930 GOSUB 250:GOSUB 280:GOSUB 290:GOSUB
210:GD=2:RETURN
940 X=X+4:IF X>174 THEN X=174:GOSUB 250
:GOSUB 210:GD=2:RETURN
950 GOSUB 250:GOSUB 240:RETURN
960 GOSUB 250:GOSUB 210:GD=2:GOSUB 2500
:RETURN
980 GOSUB 250:GOSUB 210:GOSUB 230:GOSUB
220:GD=1:RETURN
2500 REM ** SHOT **
2510 FOR I=140 TO 30 STEP -10
2520 GOSUB 2900
2530 NEXT
2540 RETURN
2900 REM ** SOUND **
2910 SOUND2:SOUND@15,I/5
2920 LINE[2]X+6,I-10,X+6,I:LINE[0]X+6,I
-10,X+6,I
```

## GUNDAM-2の解説

GUNDAM-1をカセットにセーブしたらNEW [RETURN] と  
してGUNDAM-2のプログラムを打ち込みます。

GUNDAM-2は大ザクと小ザクを画面に表示し、自動的にそれぞれ移動させるものです。

プログラムの行番号と内容はそれぞれ次のようになります。

600～646 メイン・ルーチン

1000～1070 大ザクと小ザクを画面に表示するサブ・  
ルーチン

4430～4500 大ザクと小ザクの画面データとその読み  
込み

プログラムをRUNさせて大ザクと小ザクが画面に表われ  
動いたら、SAVE "GUNDAM-2" [RETURN] でカセット  
テープにセーブします。

## GUNDAM-3の解説

NEW [RETURN] を入力してください。

GUNDAM-3は、これだけでは何の意味もありません。

GUNDAM-1とGUNDAM-2のプログラムをつなぎあわせて  
ガンドムとザクが対決するゲームになるようにします。

プログラムの行番号と内容はそれぞれ次のようになります。

610～646 メイン・ルーチンの付けたし部分

2503～2506 2600～2750 ザクに弾が当たった場合の  
サブ・ルーチン

3000～3120 制限時間を越した時の部分

4350～4420 ザクが爆発した時のマークとデータとそ  
の読み込み

このプログラムをSAVE "GUNDAM-3" [RETURN] でカ  
セットテープにセーブします。

この3つのプログラムでガンドムをするには以下のよう  
にします。

LOAD "GUNDAM-1" [RETURN]

APPEND "GUNDAM-2" [RETURN]

APPEND "GUNDAM-3" [RETURN]

そしてRUN [RETURN] でゲームができます。このゲームは  
一定時間内にできるだけ多くの大ザクと小ザクをガンドム  
のビームライフルで打ち落とすゲームです。大ザクは  
10点小ザクは20点となっています。

最後に全体のプログラムを示します。

```

2930 SOUND:RETURN
4120 FOR I=1 TO 21
4130 READ B:GA$=GA$+CHR$(B)
4140 NEXT
4150 DATA 64,64,64,192,65,82,110,126,12
7,59,59,59,127,111,110,108,108,108,1
08,238
4160 FOR I=1 TO 16
4170 READ B:R1$=R1$+CHR$(B)
4180 NEXT
4190 DATA 144,88,58,44,124,92,204,140,1
56,52,118,102,99,99,99,227
4200 FOR I=1 TO 34
4210 READ B:RR$=RR$+CHR$(B)
4220 NEXT
4230 DATA 0,0,0,5,0,109,64,63,128,13,19
2,15,192,15,128,13,192,13,192,5,96,1,48,
1
4240 DATA 248,1,236,0,44,0,4,0,0,0
4250 FOR I=1 TO 16
4260 READ B:L1$=L1$+CHR$(B)
4270 NEXT
4280 DATA 9,26,92,52,62,58,51,49,57,44,
110,102,198,198,198,199
4300 FOR I=1 TO 34
4310 READ B:LL$=LL$+CHR$(B)
4320 NEXT
4330 DATA 0,0,160,0,182,2,252,1,176,3,2
40,3,240,1,176,3,176,3,160,6,128,12,128,
31
4340 DATA 128,55,0,52,0,0,0,0,0,0
4510 GOTO 600

```

## GUNDAM-2

```

110 PRINT CHR$(6)
120 GOTO 4430
600 REM ** MAIN **
610 CT=0
643 GOSUB 1000
645 IF CT>500 THEN END
646 GOTO 643
1000 REM ** ZAG **
1005 SOUND1,3:SOUND@10,X2
1010 POSITION X3,Y3: PATTERN[0]1,ZZ$
1020 POSITION X0,Y0: PATTERN[0]1,ZAG$
1030 X3=X2: Y3=Y2: POSITION X2,Y2: PATTERN
[3]1,ZZ$
1040 X0=X1: Y0=Y1: POSITION X1,Y1: PATTERN
[2]1,ZAG$
1050 X2=-SIN(CT/30)*(RND(1)*30+40)+90: Y
2=-COS(CT/10)*25+60
1060 X1=SIN(CT/30)*(RND(1)*30+40)+90: Y1
=COS(CT/10)*25+60
1070 RETURN
4430 FOR I=1 TO 6
4440 READ B:ZAG$=ZAG$+CHR$(B)
4450 NEXT
4460 DATA 72,86,116,92,20,4
4470 FOR I=1 TO 13
4480 READ B:ZZ$=ZZ$+CHR$(B)
4490 NEXT
4500 DATA 208,172,252,52,248,56,108,108,
108,108,108,238,0
4510 GOTO 600

```

\* サンプルプログラム中に出てくる〔 〕はキーボード上で（ ）と同じです。

### GUNDAM-3

```
120 GOTO 4120
610 X=90:GD=2:CT=0
620 GOSUB 3110
645 IF CT>500 THEN 3000
646 GOTO 630
2503 IF (X3<X+6)*(X<X3+1) THEN 2600
2506 IF (X0<X+6)*(X<X0+1) THEN 2700
2600 IF (X0<X+6)*(X<X0+1)*(Y0>Y3) THEN
2700
2610 FOR I=140 TO Y3 STEP -10:GOSUB 290
0:NEXT
2620 POSITION X3,Y3: PATTERN[0]1,ZZ$
2630 POSITION X3,Y3: PATTERN[5]1,B1$:FOR
I=0TO100:NEXT:POSITION X3,Y3: PATTERN[0]1
,B1$
2640 POSITION X3,Y3: PATTERN[6]1,B2$:FOR
I=0TO100:NEXT:POSITION X3,Y3: PATTERN[0]1
,B2$
2650 SC=SC+10:CURSOR20,1:PRINT USING "#"
###";SC:GOTO 2750
2700 FOR I=140 TO Y0 STEP -10:GOSUB 290
0:NEXT
2710 POSITION X0,Y0: PATTERN[0]1,ZAG$
2720 POSITION X0,Y0: PATTERN[5]1,B1$:FOR
I=0TO100:NEXT:POSITION X0,Y0: PATTERN[0]1
,B1$
2730 POSITION X0,Y0: PATTERN[6]1,B2$:FOR
I=0TO100:NEXT:POSITION X0,Y0: PATTERN[0]1
,B2$
2740 SC=SC+20:CURSOR20,1:PRINT USING "#"
###";SC
2750 FOR I=50TO90STEP5:SOUND0,3:SOUND@15
,I/5:FOR J=0TO9:NEXT:SOUND:NEXT:RETURN
3000 REM ** PLAY AGAIN **
3010 SOUND:CURSOR 3,10:PRINT "PLAY AGAI
N (Y/N)":GETQ$
3020 IF Q$="Y" GOSUB 3100:SC=0:GOTO 600
3030 IF Q$="N" GOSUB 3100:END
3040 GOTO 3010
3100 IF SC>HS THEN HS=SC
3105 PRINT CHR$(6)
3110 CURSOR1,1:PRINT"HI-SCORE";:PRINT U
SING"####HS;:PRINT" SCORE";:PRINTUSING
"####";SC
3120 RETURN
4350 FOR I=1 TO 7
4360 READ B:B1$=B1$+CHR$(B)
4370 NEXT
4380 DATA 146,68,16,170,16,68,146
4390 FOR I=1 TO 7
4400 READ B:B2$=B2$+CHR$(B)
4410 NEXT
4420 DATA 0,0,16,40,16,0,0
4510 GOTO 600
```

### GUNDAM

```
110 PRINT CHR$(6)
120 GOTO 4120
200 POSITION XX,150: PATTERN[0]2,LL$:RET
URN
210 POSITION X,150: PATTERN[7]1,GA$:RETU
RN
220 POSITION X,150: PATTERN[7]2,LL$:RETU
RN
230 POSITION XX,150: PATTERN[0]1,GA$:RET
URN
240 POSITION X,150: PATTERN[7]2,RR$:RETU
RN
250 POSITION XX,150: PATTERN[0]2,RR$:RET
URN
260 POSITION XX,150: PATTERN[7]1,L1$:RET
URN
270 POSITION XX,150: PATTERN[0]1,L1$:RET
URN
280 POSITION XX,150: PATTERN[7]1,R1$:RET
URN
```

```
290 POSITION XX,150: PATTERN[0]1,R1$:RET
URN
600 REM ** MAIN **
610 X=90:GD=2:CT=0
620 GOSUB 3110
625 GOSUB 210
630 USR($00D8):GETKE$:XX=X
640 ON GD GOSUB 650,750,900:GOSUB 1000
643 GOSUB 1000
645 IF CT>500 THEN 3000
646 GOTO 630
650 IF KE$="P" THEN 730
655 IF KE$=" " THEN 700
660 IF KE$="O" THEN 670
665 GOSUB 200:GOSUB 260:GOSUB 270:GOSUB
210:GD=2:RETURN
670 X=X-4:IF X<4 THEN X=4:GOSUB 200:GOS
UB 210:GD=2:RETURN
675 GOSUB 200:GOSUB 220:RETURN
700 GOSUB 200:GOSUB 210:GD=2:GOSUB 2500
:RETURN
730 GOSUB 200:GOSUB 210:GOSUB 230:GOSUB
240:GD=3:RETURN
750 IF KE$=" " THEN 2500
760 IF KE$="P" THEN 850
770 IF KE$="O" THEN 800
780 FOR I=0 TO 50:NEXT:RETURN
800 X=X-4:IF X<4 THEN X=4:GOSUB 230:GOS
UB 210:RETURN
810 GOSUB 230:GOSUB 260:GOSUB 270:GOSUB
220:GD=1:RETURN
850 X=X+4:IF X>174 THEN X=174:GOSUB 230
:GOSUB 210:RETURN
860 GOSUB 230:GOSUB 280:GOSUB 290:GOSUB
240:GD=3:RETURN
900 IF KE$="O" THEN 980
910 IF KE$=" " THEN 960
920 IF KE$="P" THEN 940
930 GOSUB 250:GOSUB 280:GOSUB 290:GOSUB
210:GD=2:RETURN
940 X=X+4:IF X>174 THEN X=174:GOSUB 250
:GOSUB 210:GD=2:RETURN
950 GOSUB 250:GOSUB 240:RETURN
960 GOSUB 250:GOSUB 210:GD=2:GOSUB 2500
:RETURN
980 GOSUB 250:GOSUB 210:GOSUB 230:GOSUB
220:GD=1:RETURN
1000 REM ** ZAG **
1005 SOUND1,3:SOUND@10,X2
1010 POSITION X3,Y3: PATTERN[0]1,ZZ$
1020 POSITION X0,Y0: PATTERN[0]1,ZAG$
1030 X3=X2:Y3=Y2:POSITION X2,Y2: PATTERN
[3]1,ZZ$
1040 X0=X1:Y0=Y1:POSITION X1,Y1: PATTERN
[2]1,ZAG$
1050 X2=-SIN(CT/30)*(RND(1)*30+40)+90:Y
2=-COS(CT/10)*25+60
1060 X1=SIN(CT/30)*(RND(1)*30+40)+90:Y1
=COS(CT/10)*25+60
1070 RETURN
2500 REM ** SHOT **
2503 IF (X3<X+6)*(X<X3+1) THEN 2600
2506 IF (X0<X+6)*(X<X0+1) THEN 2700
2510 FOR I=140 TO 30 STEP -10
2520 GOSUB 2900
2530 NEXT
2540 RETURN
2600 IF (X0<X+6)*(X<X0+1)*(Y0>Y3) THEN
2700
2610 FOR I=140 TO Y3 STEP -10:GOSUB 290
0:NEXT
2620 POSITION X3,Y3: PATTERN[0]1,ZZ$
2630 POSITION X3,Y3: PATTERN[5]1,B1$:FOR
I=0TO100:NEXT:POSITION X3,Y3: PATTERN[0]1
,B1$
2640 POSITION X3,Y3: PATTERN[6]1,B2$:FOR
I=0TO100:NEXT:POSITION X3,Y3: PATTERN[0]1
,B2$
2650 SC=SC+10:CURSOR20,1:PRINT USING "#"
###";SC:GOTO 2750
```

\* サンプルプログラム中に出てくる〔 〕はキーボード上で( )と同じです。

```

2700 FOR I=140 TO Y0 STEP -10:GOSUB 290
0:NEXT
2710 POSITION X0,Y0:PATTERN[0]1,ZAG$
2720 POSITION X0,Y0:PATTERN[5]1,B1$.FOR
I=0TO100:NEXT:POSITION X0,Y0:PATTERN[0]1
,B1$
2730 POSITION X0,Y0:PATTERN[6]1,B2$:FOR
I=0TO100:NEXT:POSITION X0,Y0:PATTERN[0]1
,B2$
2740 SC=SC+20:CURSOR20,1:PRINT USING "#"
####;SC
2750 FOR I=50TO90STEP5:SOUND0,3:SOUND@15
,I/5:FOR J=0TO9:NEXT:SOUND:NEXT:RETURN
2900 REM ** SOUND **
2910 SOUND2:SOUND@15,I/5
2920 LINE[2]X+6,I-10,X+6,I:LINE[0]X+6,I
10,X+6,I
2930 SOUND:RETURN
3000 REM ** PLAY AGAIN **
3010 SOUND:CURSOR 3,10:PRINT "PLAY AGAI
N (Y/N) ":"GETQ$"
3020 IF Q$="Y" GOSUB 3100:SC=0:GOTO 600
3030 IF Q$="N" GOSUB 3100:END
3040 GOTO 3010
3100 IF SC>HS THEN HS=SC
3105 PRINT CHR$(6)
3110 CURSOR1,1:PRINT"HI-SCORE";:PRINT U
SING"####HS";:PRINT" SCORE";:PRINT USING
"####";SC
3120 RETURN
4120 FOR I=1 TO 21
4130 READ B:GA$=GA$+CHR$(B)
4140 NEXT
4150 DATA 64,64,64,192,65,82,110,126,12
7,59,59,59,127,111,110,108,108,108,1
08,238
4160 FOR I=1 TO 16
4170 READ B:R1$=R1$+CHR$(B)
4180 NEXT

```

```

4190 DATA 144,88,58,44,124,92,204,140,1
56,52,118,102,99,99,99,227
4200 FOR I=1 TO 34
4210 READ B:RR$=RR$+CHR$(B)
4220 NEXT
4230 DATA 0,0,0,5,0,109,64,63,128,13,19
2,15,192,15,128,13,192,13,192,5,96,1,48,
1
4240 DATA 248,1,236,0,44,0,4,0,0,0
4250 FOR I=1 TO 16
4260 READ B:L1$=L1$+CHR$(B)
4270 NEXT
4280 DATA 9,26,92,52,62,58,51,49,57,44,
110,102,198,198,198,199
4300 FOR I=1 TO 34
4310 READ B:LL$=LL$+CHR$(B)
4320 NEXT
4330 DATA 0,0,160,0,182,2,252,1,176,3,2
40,3,240,1,176,3,176,3,160,6,128,12,128,
31
4340 DATA 128,55,0,52,0,0,0,0,0,0
4350 FOR I=1 TO 7
4360 READ B:B1$=B1$+CHR$(B)
4370 NEXT
4380 DATA 146,68,16,170,16,68,146
4390 FOR I=1 TO 7
4400 READ B:B2$=B2$+CHR$(B)
4410 NEXT
4420 DATA 0,0,16,40,16,0,0
4430 FOR I=1 TO 6
4440 READ B:ZAG$=ZAG$+CHR$(B)
4450 NEXT
4460 DATA 72,86,116,92,20,4
4470 FOR I=1 TO 13
4480 READ B:ZZ$=ZZ$+CHR$(B)
4490 NEXT
4500 DATA 208,172,252,52,248,56,108,108
108,108,108,238,0
4510 GOTO 600

```

アキラムの戦闘プログラムです。このプログラムは、GUNMAN2版の改良版で、より複雑な戦闘システムを実装しています。敵機の飛行ルートや攻撃範囲が複数あるため、戦闘戦略が豊富になります。また、敵機の種類も複数あります。敵機の飛行ルートや攻撃範囲が複数あるため、戦闘戦略が豊富になります。また、敵機の種類も複数あります。

\* サンプルプログラム中に出てくる [ ] はキーボード上で ( ) と同じです。

## ROBOT LANGUAGE

ロボット言語と言われているもので、図形の方向と長さを示す文字型変数を入力すると、そのデータに基づいて10個分の図形を描きます。データの入力は45度右回りをT、長さを1～9の1桁の数字で与えます。図形が画面の座標外になるか10個分のデータを描くと終りになります。データの最初は必ずTを入力してください。

入力例①? T9T9T9T1T9T1T9 RETURN

②? T9TT9TTT9TTTT9TTTTT9 RETURN

## NE-Demo-1

### ROBOT LANGUAGE

```
10 REM ** ROBOT LANGUAGE **
20 PRINT CHR$(6)
30 COLOR 7,1
40 ON ERROR GOTO 160
50 X=95:Y=90
60 INPUT T$
70 IF VAL(LEFT$(T$,1))>0 THEN 60
80 FOR I=0 TO 9
90 FOR J=1 TO LEN(T$)
100 D=VAL(MID$(T$,J,1))
110 IF D=0 THEN A=A+π/4:NEXT J,I:END
120 FOR H=1 TO D
130 X=X+SIN(A):Y=Y+COS(A)
140 SET X,Y
150 NEXT H,J,I
160 END
```

## GRAPHIC-4

四角形の中にさらに連続的に四角形を回転させて描いたものです。LINEをこのようにFOR～NEXT文の中で効果的に使うと、とてもきれいな模様を描くことができます。

## ROLLING BOX

板が空中で回転している様子を三角関数で座標計算して斜めから見ているように描いたものです。LINE文の色は乱数でそのつど求めています。時間がたつ程、カラフルな画面になります。

### GRAPHIC-4

```
10 REM ** LINE **
20 PRINT CHR$(6)
30 COLOR ,0
40 BOX[7] 40,20,145,180
50 FOR I=40 TO 145 STEP 5
60 LINE[1] 40,180+(40-I)*16/10.5,I,20
70 LINE[4] I,20,145,16/10.5*(I-40)+20
80 LINE[5] 145,16/10.5*(I-40)+20,185-I,
180
90 LINE[6] 185-I,180,40,180+(40-I)*16/1
0.5
100 NEXT I
110 END
```

### ROLLING BOX

```
10 REM ** ROLLING BOX **
20 PRINT CHR$(6)
30 Y1=92 :X1=90 :Y2=92 :X2=110
40 R1=50 :R2=90
50 FOR I=10 TO 100 STEP 10
60 S=π/I
70 FOR N=1 TO 7
80 FOR J=-π/2 TO 1.5*π STEP S
90 C1=RND(1)*6+1
100 AY=R1*SIN(J) :AX=(R1/2)*COS(J)
110 BY=R2*SIN(J) :BY=(R2/2)*COS(J)
120 LINE[C1]AX+X1,AY+Y1,BX+X1,BY+Y1,BX+
X2,BY+Y2,AX+X2,AY+Y2,AX+X1,AY+Y1
130 NEXT J,N,I
140 END
```

## LINE-Demo-1

3つのFOR～NEXT文の中で共通のサブルーチンを使って直線を引いている。直線の2点の座標はそれぞれ三角関数で計算されている。このようにサブルーチンを用いるとプログラムを短かくすることができる。

## LINE-Demo-2

FOR～NEXTの中でLINE文を用いたもので同じ図形を色を変えて同じように描いている。この時バックカラーをいろいろ変えてやってみるとよい。そうすれば、カラーコードとバックカラーの関係による違いがわかるだろう。

### LINE-Demo-1

```

10 REM *** LINE Demo. ***
20 PRINT CHR$(6) : COLOR, 7
30 FOR I= /30 TO 2* + /30 STEP /30
40 GOSUB 110 : NEXT I
50 FOR I=0 TO 2* STEP /20
60 GOSUB 110 : NEXT I
70 FOR I= /60 TO 2* + /60 STEP /20
80 GOSUB 110 : NEXT I
90 GET K$: IF K$<>" " THEN 90
100 COLOR, 0 : END
110 REM *** LINE DROW ***
120 A=100*COS(I)+100 : B=5*SIN(I)+100
130 C=100*COS(I+ /1.5)+100 : D=5*SIN(I+
/1.5)+170
140 LINE[1]A,B,C,D
150 RETURN

```

### LINE-Demo-2

```

10 REM ** LINE サンプル **
20 PRINT CHR$(6) : COLOR, 0 : COL=3
30 A=.5 : B=0: C=2*π
40 XA=96 : YA=92
50 D=12 : E=90
60 I=(C-B)/D
70 FOR T1=B TO C-.1 STEP I
80 FOR T2=0 TO T1 STEP I
90 XB=E*SIN(T1)+XA
100 YB=S*E*COS(T1)*A+YA
110 XC=E*SIN(T2)+XA
120 YC=2*E*COS(T2)*A+YA
130 LINE[COL]XB,YB,XC,YC
140 NEXT T2
150 NEXT T1
160 COL=COL+1 : IF COL=8 THEN COL=1
170 GOTO 30

```

\* サンプルプログラム中に出てくる〔 〕はキーボード上で( )と同じです。