

# MSX

Graphic Workbook

# グラフィック・ワークブック

ビジュアル体験で学ぶBASICプログラミング

桜田幸嗣・蓑島 聡 共著



アスキー出版局

目次

イントロダクション

第3章 アートへのアプローチ

直線で模様を描こう 68

Stardust Night 78

円は異なるもの味なもの 80

第4章 ゲームへのアプローチ

迷路を作る 116

音符と共に音楽を！ 118

飛行機を飛ばす 120

ininterrupt ~ ドラゴン曲線 ~ 161

奥付

付録

デザインシート

キャラクターコード表

---



Graphic Workbook

# グラフィック・ワークブック

ビジュアル体験で学ぶBASICプログラミング

桜田幸嗣・蓑島 聡 共著

アスキー出版局



本文デザイン……宇治 晶  
本文イラスト……東恩納 裕一

## イントロダクション

本書はFM-7グラフィック・ワークブックのMSX版です。グラフィックスを通して、MSXユーザーの方に中級程度のBASICプログラミングテクニックを身につけていただくという目的で書かれました。

MSXのBASICに関しては、いままでにも「MSXビギナーズBASIC」を始め、数多くの入門書が出版されています。それらによって、BASICの基礎知識を得、次なるステップへいざ踏み出そうという方にとっては、本書のプログラムを読み、かつ理解することがプログラミングの良き練習となるでしょう。掲載されたグラフィックスの数々がその興味を一段と増すことは請け合います。

ところで、MSXという統一規格がパソコンのユーザーにとって大変なメリットを持つということは、すでにいろいろな場所で言及されています。

しかし、MSXの登場をこのうえもなく喜んでいる人々がもう1種類存在している。この事実は意外と知られておりません。そのもう1つの人種とは、何をかくそうパソコンの解説書執筆者たちであります。

すこし前までは、BASIC言語の説明をするにしても、グラフィックスを説くにしても、個々のメーカーの機種に特有な機能や制限のため、意を尽くした解説を行うことはなかなか難しかったものです。

パソコン一般について話を進めてしまうと、話題が画一的で、実際のマシン操作にはあまり役立たないものになりがちです。かといって、ある機種の特徴を十分に活かす説明を行おうと思うと、こんどは世界があまりに狭くなって、他に融通のきかない知識ばかり読者の方に与えているのではないかと、という不安が生まれてきます。

その点MSXでは、このような1冊の書籍が、どのメーカーのどのMSXマシンを持っている方にも同じ条件の上に立ってお読みいただけるため、余計なことを考えずにこちらの伝えたいことをそのまま伝えられます。こうやって書き連ねたことが過不足なく読者の方々に応用していただけるわけで、筆者としてもこんなに喜ばしいことはありません。

MSX関係の解説書を執筆している人間は、多かれ少なかれ同様な感慨を持っています。その意味で、MSXの本はどれをとってもその筆者なりの自信作となっているはず。本書もその例にもれませんが、我々は自信を持ってこの本をMSXユーザーの方々に捧げます。

昭和59年4月1日

桜田 幸嗣  
葦島 聡

# MSXグラフィック・ワークブック

## 【目次】

### 1章 忘れていませんかグラフィックの約束ごと、MSXの使い方

グラフィックを始める前に 10

- 画面モード 11
- 画面モードとCOLOR命令 12

グラフィック命令に関する基本的御注意 14

- PSET, PRESETについて 15
- LINE or DRAW 15
- CIRCLEについて 16
- PAINT命令について 16
- 色と画面モード 17

キャラクタ表示命令に関する基本的御注意 19

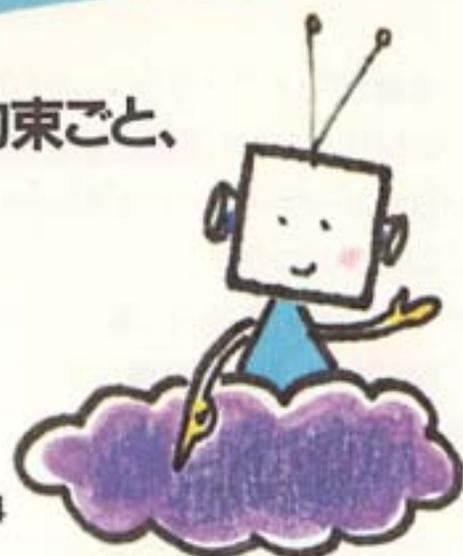
- WIDTHと趣味の画面モード 20
- グラフィックとキャラクタの混在 20
- グラフィックキャラクタについて 21

スプライト機能について 22

### 2章 グラフィックの基本操作…… 覚えてほしい、あんなこと、こんなこと、

位置を決める 30

- MSXで用いる座標 31
- 中間点を求めて 34



- 図形の移動 37
- キャラクタ座標 40

### 「絵」をつくる 41

- コマンドを並べて 42
- READ DATAの利用 43
- 角度と距離による位置の指定 44
- 乱数の利用 47

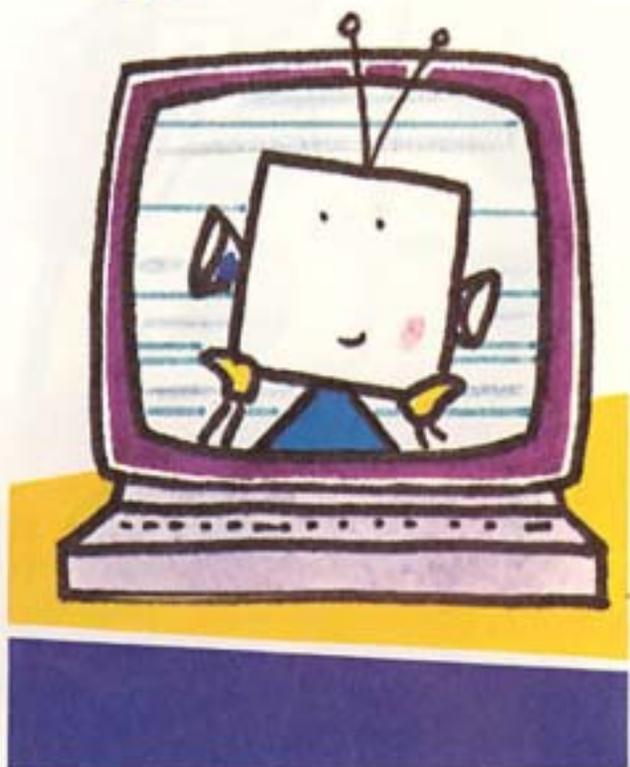
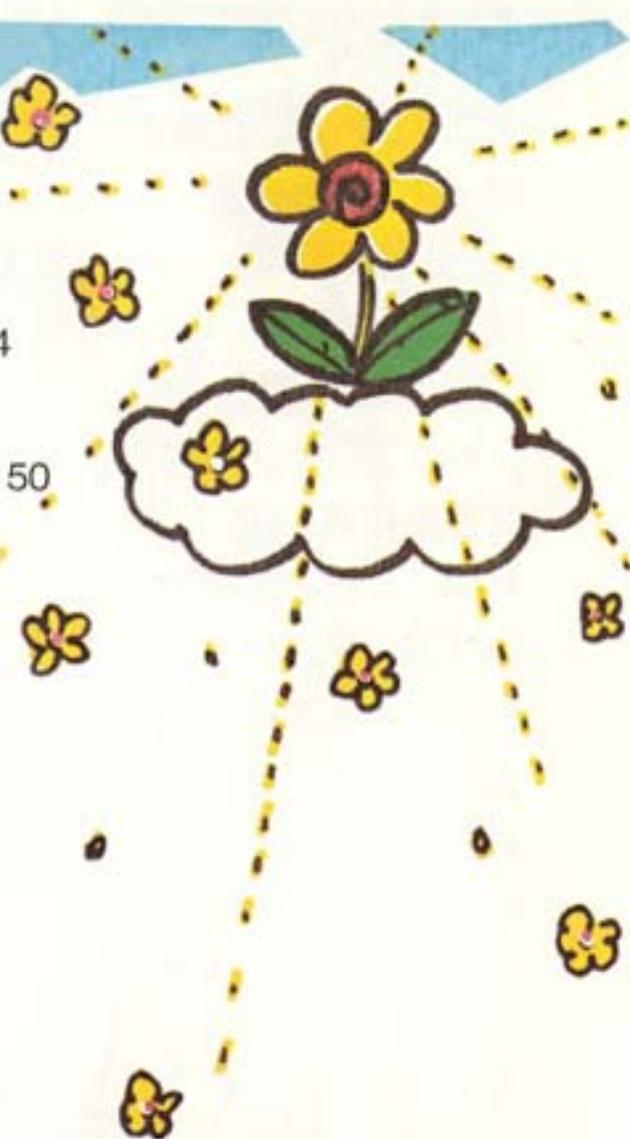
### 動くグラフィックのこころみ 50

- 同じ形を並べて 51
- 周期的繰返し 52
- 移動するグラフィック 54
- キー入力による処理 56
- 位置の判断 59
- スプライト機能を使おう 61



## 3章 アートへのアプローチ

- 直線で模様を描こう 68
- MUG CUPS 70
- スペース シャトル 72
- リンゴを描く 74
- 落葉と星型 76
- Stardust Night 78
- 円は異なるもの味なもの 80
- 自分だけの曲線を描こう 84
- ヘリコイド曲線を描く 86
- ネコでも回そうか? 88
- 3次元グラフィックス? 90
- 文字の変形 94



## 4章 ゲームへのアプローチ

- テニスゲームの仕組み 100
- ハイスコアの表示 104
- ビーム砲を打つ 106
- 基地が危ない! 108
- 図形の拡大, 縮小 110
- ストップウォッチを作る 114
- 迷路を作る 116
- 音符とともに音楽を! 118
- 飛行機を飛ばす 120
- タイマー&エネルギーメータ 122
- 流れる星空 124
- カニさんカニさん 126



## 5章 ランダムアプリケーション

バイオリズム 132

迷路ゲーム 137

焼けてないトタン屋根の上のネコ 143

スキーゲーム 153

### Appendix

三次元グラフィック・パッケージ 164

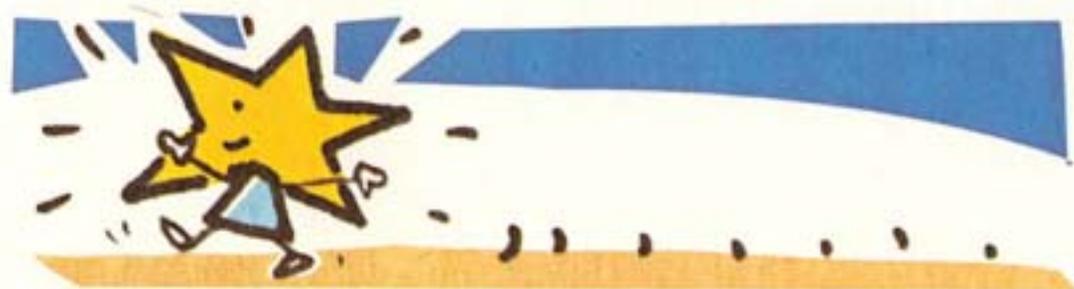
スプライトパターン・ジェネレータ 171

MSX グラフィックエディタ 176



# 【3章】

## アートへのアプローチ



直線で模様を描こう

MUG CUPS

スペースシャトル

リンゴを描く

落葉と星型

Stardust Night

円は異なるもの味なもの

自分だけの曲線を描こう

ヘリコイド曲線を描く

ネコでも回そうか?

3次元グラフィックス?

文字の変形

本章ではコンピュータによるアートに挑戦します。

近年、我等が MSX をはじめとする高性能パソコンが出現し、これまでは高嶺の花であったコンピュータ・グラフィックスが私たちにも手軽に楽しめるようになりました。なかでも、本章に掲載したような世人の目を楽ませる壮麗な、あるいは小意気に洒落たグラフィック作品が、わずかな手間で描けるこの楽しさ……。美しい作品を描くためのテクニックのその入口あたりを種々取りまぜて見てもらうことにいたします。月並ないいい方ですが、これを基礎として皆さん自身、いろいろなグラフィックに取り組んでいただきたい。筆者らはそう希望しております。

# 直線で模様を描こう

本章のテーマであるアートという言葉に何やら高尚な、そしてむずかしそうな印象を感じた方、いえいえそんなに身構える必要はありません。

何はともあれ、まず直線を一本引くことから始めましょう。それだけでも美しい模様は描けるのです。

ここに挙げた2つのプログラムを見てください。扱っているグラフィックは、どちらもLINE 命令だけで描かれています。が、ちよつとすてきな模様となっていると思いませんか？

List 1a では、横 XB から XE、縦 YB から YE の四角形の各辺上を直線の始点と終点を移動させます。190 行で引いている直線を例にとれば、始点は四角形の底辺を S 分割して、左から右に XS ドット

ずつ動いていき、終点は四角形の右辺を下から上に向けて、YS ドットずつ動いていきます(下図参照)。この始点と終点を順に結ぶと、右下の角に相当する部分が描きあがります。同様に、200~220 行までの LINE

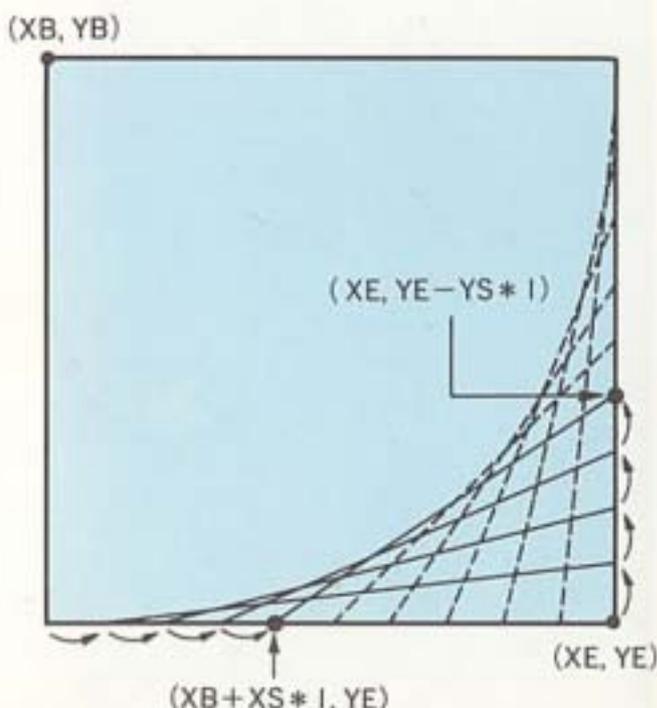
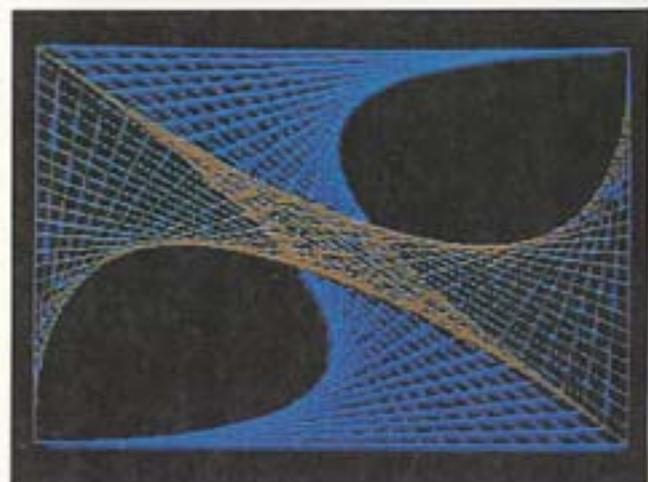


Figure 1



命令が、四角形の右上、左上、左下の角を描いていくわけです。

List 1b でも、やはり直線の始点と終点を動かしていますが、今度は四角形の辺

と対角線を結んで線を引いています。

どちらのプログラムも変数 S や XE, YE などの値を変えると画面が変わります。変化する模様を楽しんでください。

### List 1a

```

100 ' List 1a
110 '
120 XB=32 : YB=0
130 XE=223 : YE=191
140 S=36 : C=15
150 XS=(XE-XB)/S : YS=(YE-YB)/S
160 '
170 SCREEN 2 : COLOR 15,13,13 : CLS
180 FOR I=0 TO S
190 LINE (XB+XS*I, YE)-(XE, YE-YS*I), C
200 LINE (XE-XS*I, YB)-(XE, YE-YS*I), C
210 LINE (XE-XS*I, YB)-(XB, YB+YS*I), C
220 LINE (XB+XS*I, YE)-(XB, YB+YS*I), C
230 NEXT I
240 DM$=INPUT$(1):END

```

グラフィックを表示する範囲

各辺を分割する数とグラフィックの色

### List 1b

```

100 ' List 1b
110 '
120 XB=0 : YB=0
130 XE=255 : YE=191
140 S=25 : C1=4 : C2=6
150 XS=(XE-XB)/S : YS=(YE-YB)/S
160 '
170 SCREEN 2 : COLOR 7,1,1 : CLS
180 FOR I=0 TO S
190 LINE (XE-XS*I, YB)-(XB+XS*I, YB+YS*I), C1
200 LINE (XB+XS*I, YE)-(XE-XS*I, YE-YS*I), C1
210 NEXT I
220 FOR I=0 TO S
230 LINE (XE, YB+YS*I)-(XE-XS*I, YE-YS*I), C2
240 LINE (XB, YE-YS*I)-(XB+XS*I, YB+YS*I), C2
250 NEXT I
260 DM$=INPUT$(1):END

```

# Stardust Night

MSX のグラフィック画面は、256×192ドットの解像度があり、見ごたえのある美しい絵を描くことも十分可能です。ここではグラフィックのサンプル・プログラムとしてはスタンダードな、スターダストナイトに挑戦してみましょう。

ここで、プログラムを見てください。まず、バックに散りばめられる星は270~320行で表示しています。280、290行で乱数を発生させ、この位置に星を表示します。ただし、星の分布を Y 座標の小さい方、つまり画面の上の方へ寄せるため、星の Y 座標は290行のように、 $RND(1) * RND(1)$ によ

る乱数を用います。

次に流星ですが、これは400~490行のサブルーチンで描いています。このサブルーチンは、370行の  $RND(1) < .002$  という条件が満たされたときにのみ呼び出し実行されます。要するに370行を実行するたびに、星を描くと流れ星の大安売りとなってしまうのを防いだのです。派手好きな人、さびしがりの人は、370行を、ただの GOSUB 400とし、実行してください。流星自体は乱数で始点、終点を決め、PUT SPRITE 命令で描いています。



## List 6

```

100 ' List 6
110 '
120 RESTORE 510 : ST$=''
130 FOR I=1 TO 32
140   READ D : ST$=ST$+CHR$(D)
150 NEXT I
160 '
170 SCREEN 2,2 : COLOR 15,4,1 : CLS
180 DRAW 'S4A0 BM160,191 C1'
190 DRAW 'M160,178 M170,170 M207,170'
200 DRAW 'M207,158 M216,152 M224,158'
210 DRAW 'M224,170 M234,178 M234,191'
220 DRAW 'BM216,152 M216,140'
230 DRAW 'BM212,146 M220,146'
240 PAINT (216,160),1
250 LINE (212,160)-(218,168),8,BF
260 '
270 FOR I=1 TO 200
280   X=INT(RND(1)*32)*8
290   Y=INT(RND(1)*RND(1)*96)*2
300   C=INT(RND(1)*14)+2
310   PSET (X,Y),C
320 NEXT I
330 '
340 CIRCLE (75,80),8,11 : PAINT (75,80),11 .....月を描く
350 '
360 SPRITE$(0)=ST$
370 IF RND(1)<.002 THEN GOSUB 400 .....流星を描くサブルーチンを呼ぶ
380 IF INKEY$<>' THEN END ELSE 370
390 '
400 X0=INT(RND(1)*100)+20 : Y0=INT(RND(1)*30) : Y=Y0
410 X1=INT(RND(1)*100)+130 : Y1=INT(RND(1)*70)+60
420 YS=(Y1-Y0)/(X1-X0)*4
430 PLAY 'T158L64N80N79N78N77N76N75N74'
440 FOR X=X0 TO X1 STEP 4
450   PUT SPRITE 0,(X,Y),11,0
460   Y=Y+YS
470 NEXT X
480 PUT SPRITE 0,(X,209),11,0
490 RETURN
500 '
510 DATA 128,64,32,16,8,4,2,1
520 DATA 0,0,0,1,0,0,0,0
530 DATA 0,0,0,0,0,0,0,16
540 DATA 146,84,56,255,56,84,146,16

```

星のsprayパターンを文字列として定義する

教会を描く

星を描く

月を描く

流星を描くサブルーチンを呼ぶ

星を動かすサブルーチン

流星のsprayパターン・データ

# 円は異なるもの味なもの

円を基本とした模様を描く際ひんばんに登場するのが、例のサインやコサインという三角関数です。

Figure 1 のような直角三角形において、角  $T$  と向かい合う位置の辺の長さが  $\sin(T)$  に相当し、角  $T$  と直角を結ぶ辺の長さが  $\cos(T)$  に相当する、この位置関係を三角形がどんな向きに傾いても見分ける眼力が必要です。

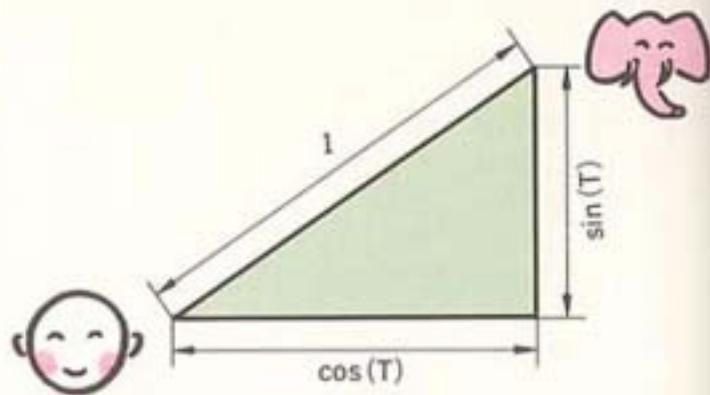
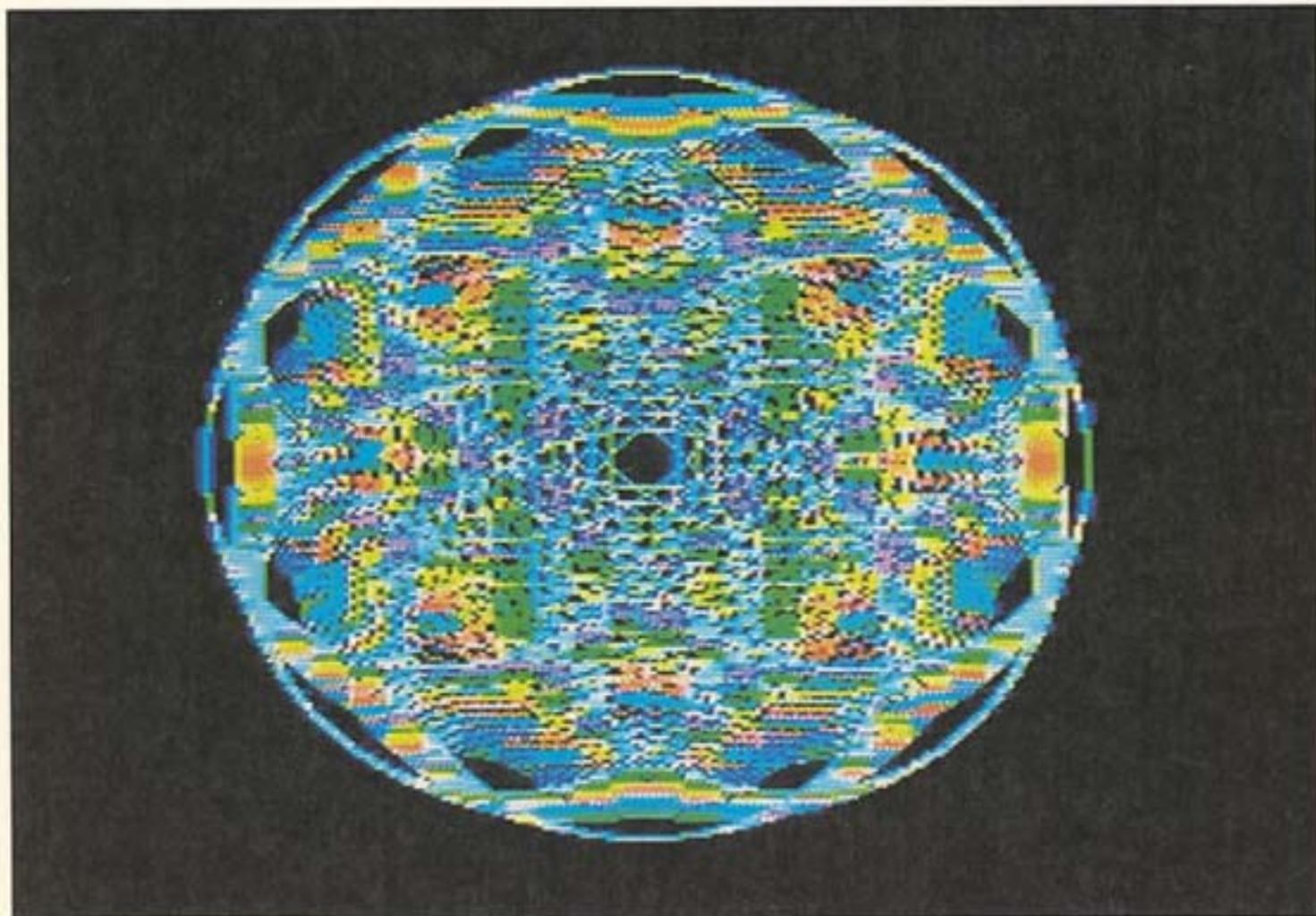
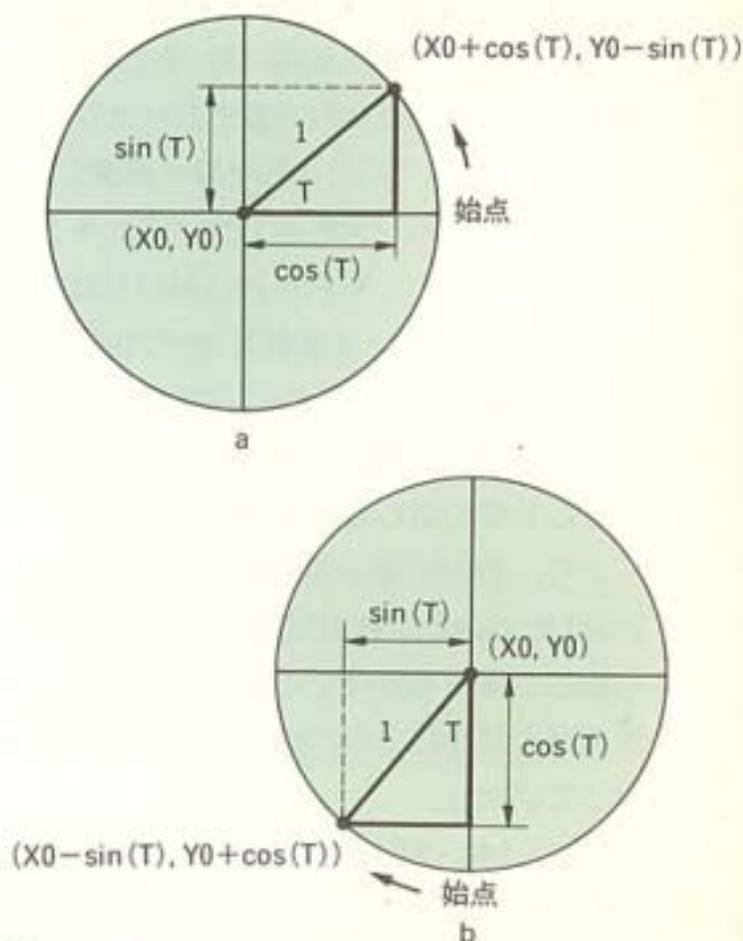


Figure 1

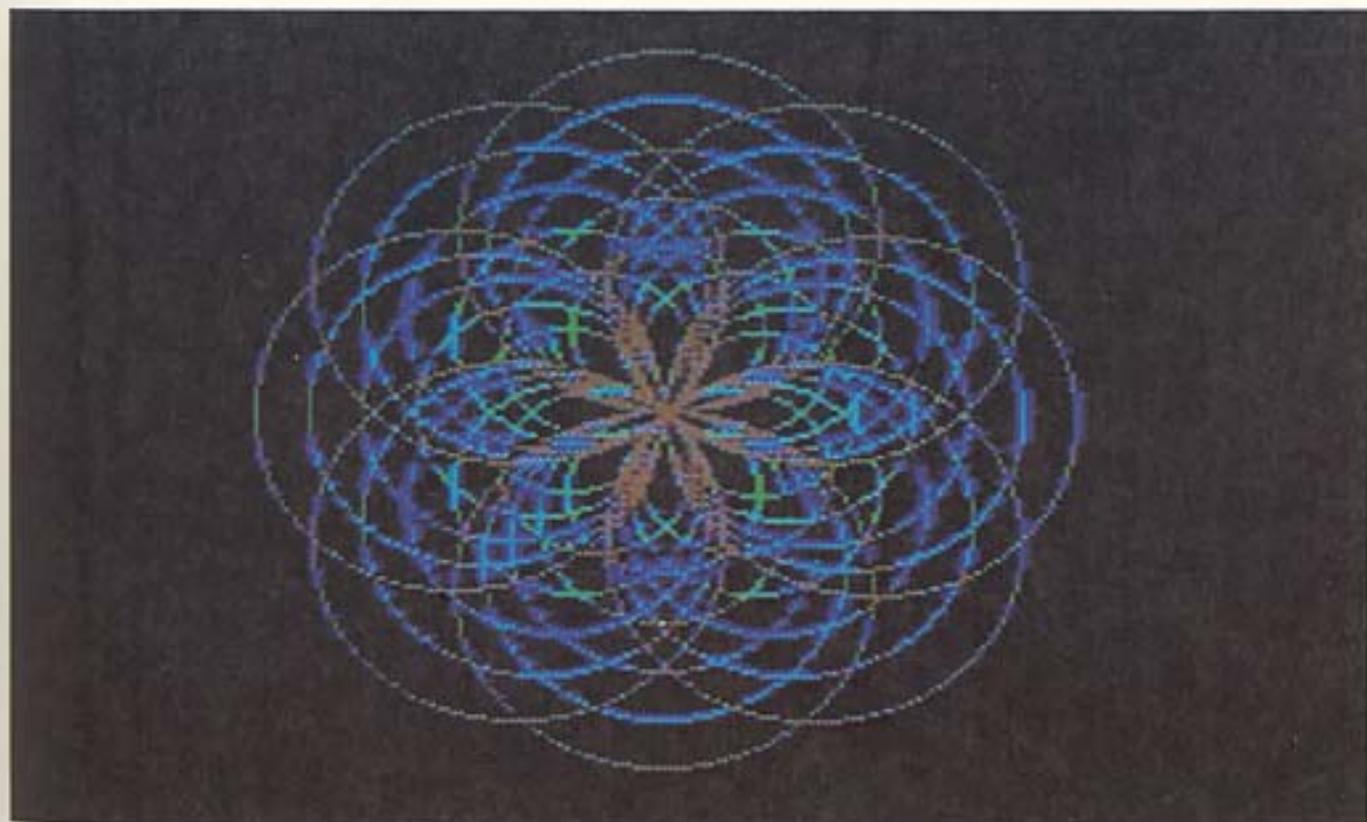


ここで *Figure 2* を見てください。始点から角度  $T$  だけ回転した円周上の位置は、 $\sin$ 関数と  $\cos$ 関数を使って計算できます。この位置は、始点の位置と角  $T$  の向きによって様々な場合がありますが、ここでは2通りの例を示しました。どちらの場合も、画面上の  $x$  座標と  $y$  座標はそれぞれ右と下へ行くほど増加するということと *Figure 1* の関係を考え合わせれば、答が出てきます。

円の中心を  $(X_0, Y_0)$  とすれば、*Figure 2a* のように、円の右端から反時計回りに角  $T$  だけ回転した位置は  $(X_0 + \cos(T), Y_0 - \sin(T))$ 、また、*Figure 2b* のように、円の下端から時計回りに回転させる場合は  $(X_0 - \sin(T), Y_0 + \cos(T))$  というように位置の計算ができるわけです。



*Figure 2*



List 7 では、上の例の Figure 2a と同じ角度のとり方を用いて、半径  $R$  の円周上に次々と小さな円を描き並べます。そこで200行のようにこの円を描く座標には、半径  $R$  を掛け、 $(X0 + \cos(T) * R, Y0 - \sin(T) * R)$  としなければなりません(Figure 3 参照)。また、180~240 行の FOR~NEXT ループで  $R$  の値を  $RB$  から  $RE$  まで次第に大きくしながら、同様なことを繰り返していきます。

さて、今のはほんの小手調べです。次に210行を List 7a のように変更し、どんな図形が現れるか見てください。これは  $R$  が大きくなるにつれ、描かれる円も大きくなっています。

では、140, 210行を List 7b のように

変えてみましょう。今度は、半径96の大きな円周の内側に接して、円が並びます。

このように、円の半径の指定を変えると、いろいろな変化が楽しめます。さらに、ここで SCREEN 3 のマルチカラーモードを使ってみました。List 7c を追加してみてください。

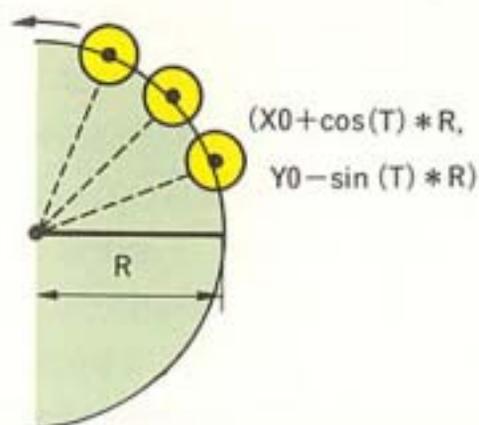
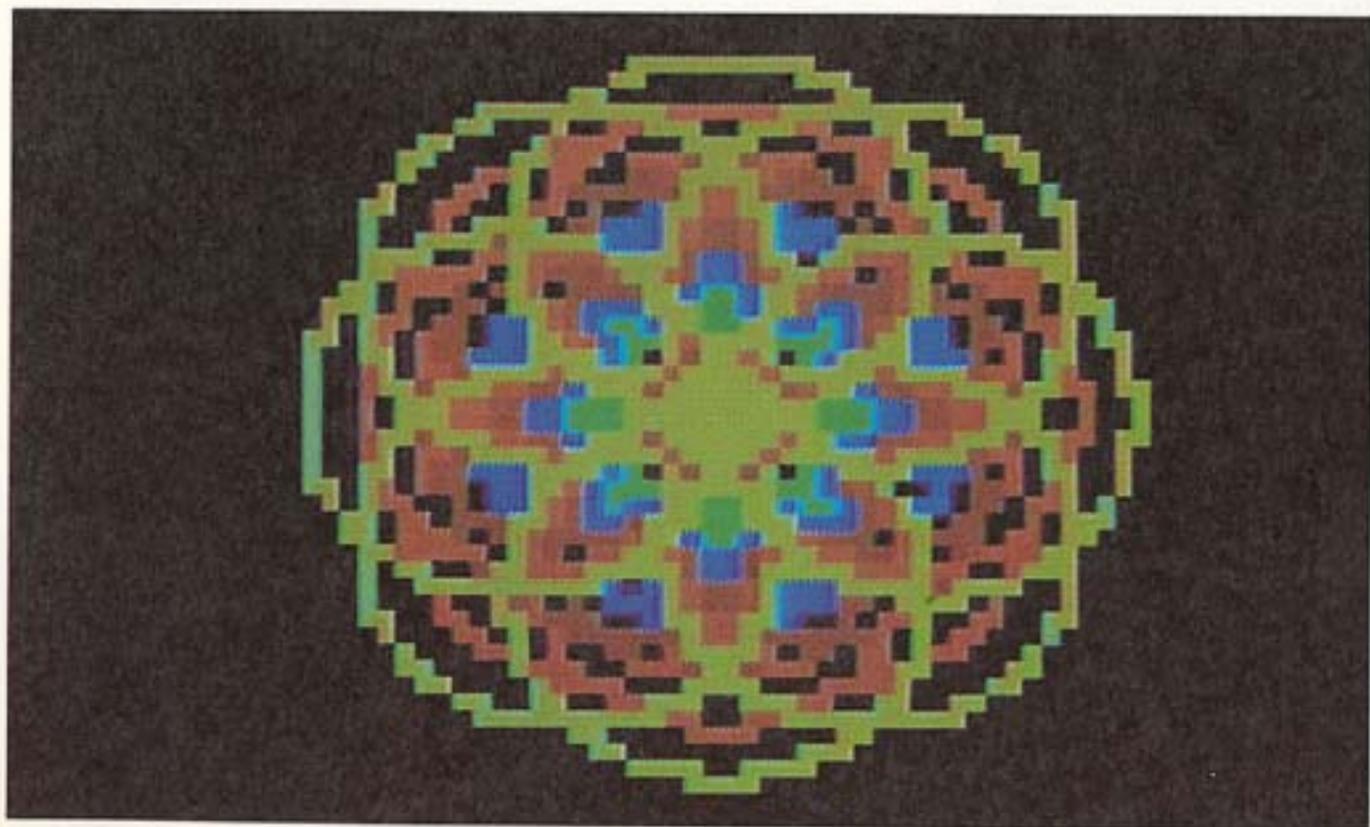


Figure 3



## List 7

```

100 ' List 7
110 '
120 DEFSNG A-Z
130 PI=3.14159 .....定数の設定
140 RB=20 : RE=44 : RS=6
150 S=8 : C=2
160 ' .....円周を8分割する
170 SCREEN 2 : COLOR 14,1,1 : CLS
180 FOR R=RB TO RE STEP RS
190   FOR T=0 TO 2*PI STEP 2*PI/S
200     X=128+R*COS(T) : Y=96-R*SIN(T)
210     CIRCLE (X,Y),4,C
220   NEXT T
230   C=C+1 : IF C=16 THEN C=2
240 NEXT R
250 DM$=INPUT$(1):END

```

## List 7a

```

210   CIRCLE (X,Y),R,C

```

## List 7b

```

140 RB=20 : RE=44 : RS=2
210   CIRCLE (X,Y),96-R,C

```

## List 7c

```

140 RB=20 : RE=44 : RS=6
150 S=8 : C=2
170 SCREEN 3 : COLOR 14,1,1 : CLS
210   CIRCLE (X,Y),R,C
230   C=C+2 : IF C>16 THEN C=2

```

# 【4章】

## ゲームへのアプローチ



テニスゲームの仕組み  
ハイスコアの表示  
ビーム砲を打つ  
基地が危ない!  
図形の拡大, 縮小  
ストップウォッチを作る  
迷路を作る  
音符とともに音楽を!  
飛行機を飛ばす  
タイマー&エネルギーメータ  
流れる星空  
カニさんカニさん

本章では, BASICプログラムによるTVゲームの可能性を追求してみたいと思います。

前章では画面に描かれたアートを見て「美しい……」と感心するのが主たる目的でした。しかし単なる静止画面のきれいさだけではTVゲームはできません。加えて動きの面白さが必要になってくるのです。グラフィックに動きを与えること, これが本章の目的です。

# 迷路を作る

誰でも、紙の上に書かれた迷路を鉛筆の先でたどっていく、という経験が、一度や二度はあると思います。

この迷路をコンピュータに作らせるにはどうすればよいのでしょうか。

迷路を作るには、道をのばしていく方法と、壁をのばしていく方法がありますが、ここでは壁をのばしていく方法を使います。Figure 1 のように入口と出口によって、

外壁は2つに分けられます。この外壁(壁a, b)に接する壁を次々に作っていくのですが、すでに存在している壁と壁をつなぐような壁を作ってははいけません。抜け道がふさがってしまうことがあるからです。この原則さえ守って壁を作っていけば、かならず抜け道のある迷路ができあがります。

このプログラムでは、220行で外壁を描き、230, 240行で入口と出口を作っています。

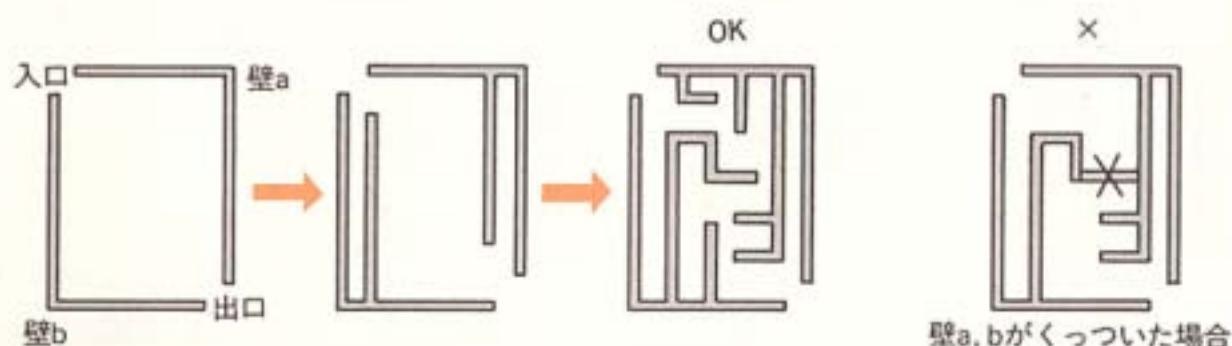
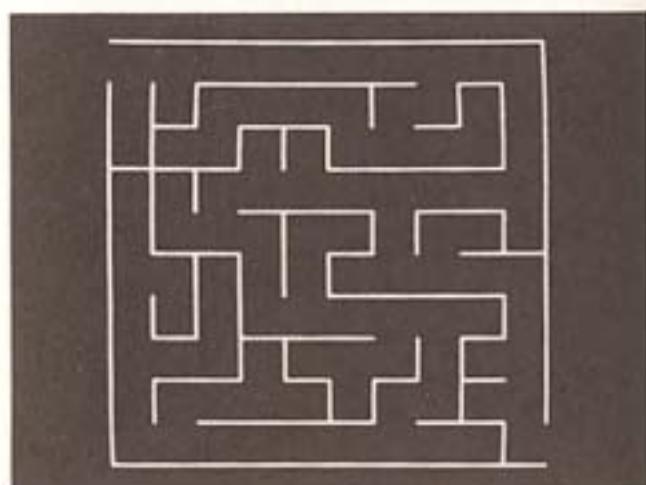
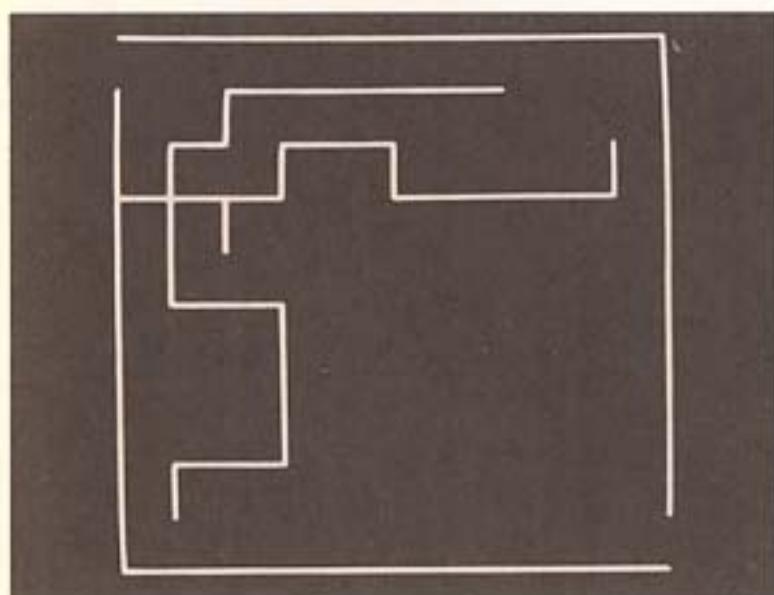


Figure 1



260~280行は、2つの壁のどちらか一方に接する点を探しています。その点が決まると、300行で発生させた乱数により、どちら向きの壁を作るか決定し、310~380行でその壁が他の壁と接しないかどうか判定し、390行

で壁を描きます。

この方法は、大きい迷路を作るとき非常に時間がかかるのが欠点です。他にも迷路の描き方はありますから、研究してみてください。

### List 7

```

100 ' List 7
110 '
120 DEFSNG A-Z
130 XB=50 : XE=200
140 YB=10 : YE=180
150 M=10 : N=10
160 XW=(XE-XB)/M : YW=(YE-YB)/N
170 S=(M-1)*(N-1)
180 DM=RND(-TIME)
190 C=7
200 '
210 SCREEN 2 : COLOR 15,1,1 : CLS
220 LINE (XB,YB)-(XE,YE),C,B .....外壁を描く
230 LINE (XB,YB+1)-(XB,YB+YW-1),1
240 LINE (XE,YE-1)-(XE,YE-YW+1),1 .....出入口を描く
250 '
260 X=INT(RND(1)*(M+1)) : Y=INT(RND(1)*(N+1))
270 XO=XB+X*XW : YO=YB+Y*YW
280 IF POINT(XO,YO)=1 THEN 260
290 FOR I=1 TO 20
300 R=INT(RND(1)*4) .....壁を延ばす方向を決める
310 IF R=0 THEN XS=-1 : YS=0 : GOTO 350
320 IF R=1 THEN XS=1 : YS=0 : GOTO 350
330 IF R=2 THEN XS=0 : YS=1 : GOTO 350
340 IF R=3 THEN XS=0 : YS=-1
350 XN=XO+XS*XW : YN=YO+YS*YW
360 IF XN<=XB OR XN>=XE THEN 410
370 IF YN<=YB OR YN>=YE THEN 410
380 IF POINT(XN,YN)<>1 THEN 410
390 LINE (XO,YO)-(XN,YN),C
400 XO=XN : YO=YN : S=S-1
410 NEXT I
420 IF S>0 THEN 260
430 PLAY *V12T240L6405A*
440 DM$=INPUT$(1):END

```

新たに作る壁が他の壁に接しないかどうか判定する

# 音符とともに音楽を!

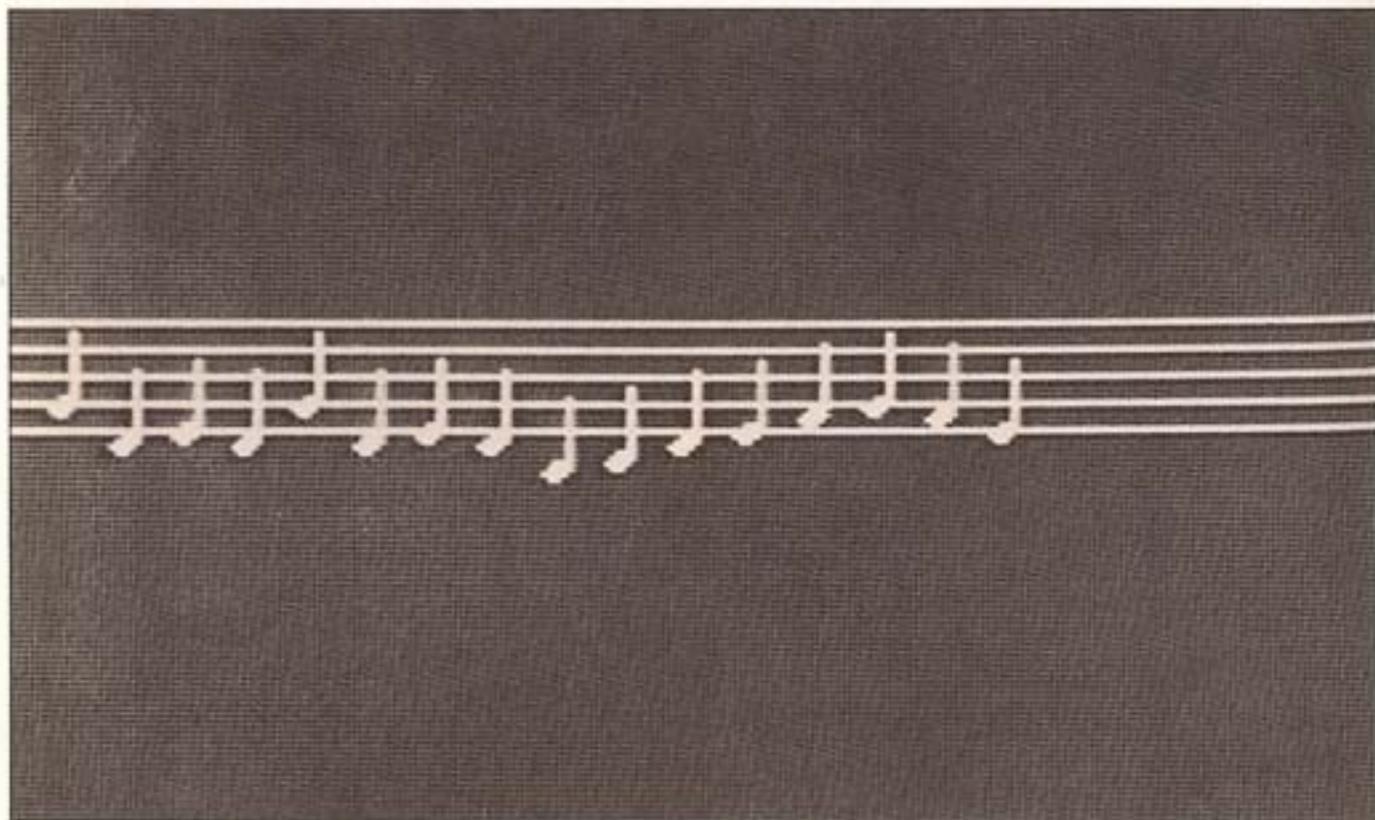
MSX は、PLAY 命令を使うことによって、音楽を演奏することができます。そこで、この音楽とともに、画面に音符を出力するプログラムを考えてみましょう。プログラムを簡単にするために、単音で音符を出力します。

音符の形 (♪) は、DRAW 命令で毎回描いています。

では、リストを見てください。音符は1画面に24個しか並べられないので、それを変数 N で数えておき、N=25になったら画面をクリアして再び左端から音符を出力しています。

音のデータを DATA として蓄えておく方法はいろいろあると思いますが、ここでは、1つひとつをカンマで区切って、大文字だけで表現させています。また、音程のデータである "C, D, E……" が、ドである "C" から数えて何番目の高さにあるかということは、130行で文字列 TNS="CDEFGAB" と定義し、270行で INSTR 関数によって得ています。この値を使って280行の式で計算し、音符を DRAW 命令で出力する場所を決めているのです。

このプログラムでは、最も単純な場合を例にとりましたが、MSX は3重和音まで出



せるので、それを音符とともに出力するよ  
うなプログラムを考えてみてください。ま

だ、休符や、#、bなども出力するとより楽  
しいものとなるでしょう。

### List 8

```

100 ' List 8
110 '
120 C=15
130 TN$='CDEFGAB'
140 PLAY 'S0M5000T80'
150 '
160 SCREEN 2 : COLOR 15,4,4 : CLS
170 GOSUB 330
180 RESTORE 420
190 N=1
200 '
210 READ A$
220 IF A$='@' THEN DM$=INPUT$(1):END
230 B$=LEFT$(A$,1)
240 IF B$='O' THEN PLAY A$ : O=VAL(RIGHT$(A$,1)) : GOTO 210
250 IF B$='L' THEN PLAY A$ : GOTO 210
260 IF B$='R' THEN PLAY A$ : GOTO 210
270 I=INSTR(TN$,A$)
280 X=N*10 : Y=105-(0-4)*17.5-(I-1)*2.5
290 PLAY A$ : GOSUB 380
300 N=N+1 : IF N=25 THEN CLS : N=1 : GOSUB 330
310 GOTO 210
320 '
330 FOR L=80 TO 100 STEP 5
340 LINE (0,L)-(255,L),15
350 NEXT L
360 RETURN
370 '
380 DRAW 'S4A0BM=X; ,=Y;BM+2,+2C15'
390 DRAW 'L1H1R3E1L3E1R2U12'
400 RETURN
410 '
420 DATA L16,04,G,D,E,D,G,D,E,D
430 DATA L32,03,B,04,C,D,E,F,G,F,E
440 DATA F,E,D,E,D,C,03,B,04,C
450 DATA 03,B,04,C,D,E,F,G,L16,03,G
460 DATA L16,04,G,D,E,D
470 DATA L32,03,B,04,C,D,E,F,G,F,E
480 DATA F,E,D,E,D,C,03,B,04,C
490 DATA 03,B,04,C,D,E,F,G,L4,03,G
500 DATA @

```

五線を描くサブルーチン

音符を描くサブルーチン

音のデータ

# 飛行機を飛ばす

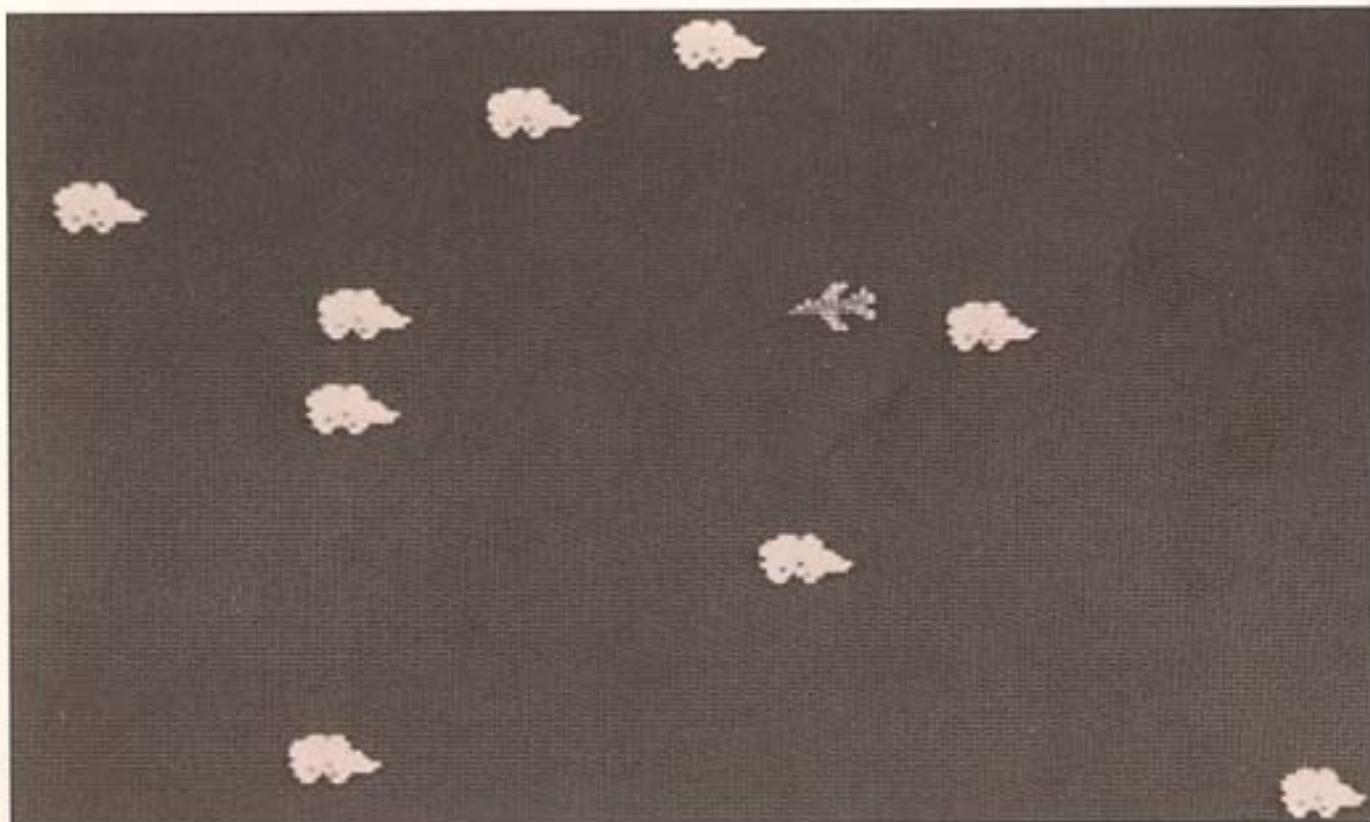
複雑な図形を素早く簡単に移動させるには、スプライト機能を使うと便利です。この機能を用いて、飛行機を飛ばしてみましょう。

飛行機は2色で表現したいので、スプライトパターンはそれぞれの色の部分について2つ作り、重ね合わせて出力しています。この2つのスプライトパターンは、スプライト面の10番と11番に割り当てました。雲は、270~300行で飛行機の割り当てられているスプライト面の前後に置き、飛行機が見え隠れするようにしています。

330~420行は、カーソルキー入力に対する基本的なテクニックです。カーソルキー $\leftarrow$ を押すと飛行機は上へ、 $\rightarrow$ を押すと下へ動いて行きます。330行では、スプライトが重なったときの割り込みを設定しています。そこで雲と飛行機が重なった場合は、440行のPLAY命令により雲との衝突音 $\text{P}$ を発します。

このプログラムは、ちょっと手を加えるだけで、ゲームになりそうですね。

なお、プログラムを終了するには、 $\text{P}$ キーを押してください。



## List 9

```

100 ' List 9
110 '
120 DEFINT A-Z
130 DM=RND(-TIME)
140 XB=0 : YB=-1
150 XS=8 : YS=4
160 XE=255-16 : YE=191-16
170 '
180 SCREEN 2,2,0 : COLOR 15,4,4 : CLS
190 RESTORE 460
200 FOR I=0 TO 2
210   S$=""
220   FOR J=1 TO 32
230     READ R : S$=S$+CHR$(R)
240   NEXT J
250   SPRITE$(I)=S$
260 NEXT I
270 FOR I=1 TO 20 STEP 2
280   X=(XE-XB)*RND(1)+XB : Y=(YE-YB)*RND(1)+YB
290   IF I<>11 THEN PUT SPRITE I,(X,Y),15,2
300 NEXT I
310 X=XE : Y=(YB+YE)/2
320 '
330 ON SPRITE GOSUB 440
340 SPRITE ON
350 I=STICK(0)
360 X=X-XS : IF X<XB THEN X=XE
370 IF I=1 AND Y-YB>YS THEN Y=Y-YS
380 IF I=5 AND Y+YS<YE THEN Y=Y+YS
390 IF I=3 OR I=7 THEN END
400 PUT SPRITE 10,(X,Y),9,0 : PUT SPRITE 11,(X,Y),11,1
410 FOR T=1 TO 20 : NEXT T
420 GOTO 340
430 '
440 SPRITE OFF : PLAY "V12L6403A" : RETURN
450 ' 雲のスプライトパターン・データ
460 DATA 0,0,0,0,0,0,0,0,31,63,124,0,0,0,0,0
470 DATA 0,0,0,0,0,0,12,28,252,224,112,0,0,0,0,0
480 ' 飛行機のスプライトパターン・データ
490 DATA 0,0,0,0,0,0,1,3,0,0,3,3,1,0,0,0
500 DATA 0,0,0,0,0,224,192,130,2,28,142,134,192,224,0,0
510 ' 雲のスプライトパターン・データ
520 DATA 0,0,14,63,127,127,63,126,55,61,24,0,0,0,0,0
530 DATA 0,0,192,224,224,248,252,255,254,96,192,0,0,0,0,0

```



スプライトパターンの設定

キー入力によって飛行機を動かすルーチン

## ドラゴン曲線

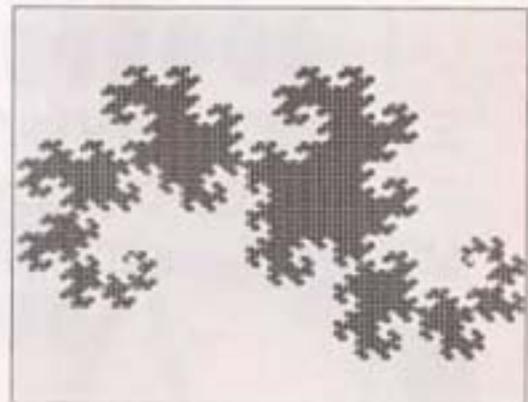
97ページで再帰的なグラフィック図形の例をひとつご紹介しましたが、かのプログラムは、サブルーチンが自分自身のみをコールするという構造をとっているため、再帰といってもまだ理解しやすい部類に入ります。けれども世の中には、あるサブルーチンが別のサブルーチンをコールし、そのサブルーチンが別のサブルーチンをコールし、巡りめぐってまた自分自身が使われるという、ほとんど人間の理解のレベルを超越した構造の再帰的グラフィックが存在しています。

ここに掲載したのも、そのようなプログラムのひとつです。もっぱら左へ曲がりたがるサブルーチン(200行)と、逆に右へばかり曲がろうとするサブルーチン(280行)が、お互いに相手をコールしながらもつれあい、ある曲線を織り上げていきます。しかし見て頂けるとおわかりのように、これだけ複雑でありながら、同じ線の上を二度通ることは絶対にありません。この曲線はドラゴンカーブと呼ばれ、この手のグラフィックが好きな人々の間ではとても有名なシロモノです。

```

100 ' List Ex.5
110 '
120 SCREEN 2 : COLOR 1,15,15 : CLS
130 DEFINT A-Z
140 '
150 C=1
160 DRAW 'BM70,120 A1S8 C=C;'
170   R=1 : D=12 : GOSUB 200 : C=16-C
180 GOTO 160
190 '
200 IF D=0 THEN DRAW 'U' : RETURN
210   D=D-1
220   GOSUB 200
230   R=(R+3) MOD 4 : DRAW 'A=R;'
240   GOSUB 280
250   D=D+1
260 RETURN
270 '
280 IF D=0 THEN DRAW 'U' : RETURN
290   D=D-1
300   GOSUB 200
310   R=(R+1) MOD 4 : DRAW 'A=R;'
320   GOSUB 280
330   D=D+1
340 RETURN

```



## 参考文献

- [1] 「PC-8801 グラフィックスのすべて」  
工藤丈彦 他  
アスキー出版局
- [2] 「PC-9801 ガイドブック」  
榊 正憲 他  
アスキー出版局
- [3] 「コンピュータ グラフィックス」  
DAVID F. ROGERS  
J. ALAN ADAMS  
山口富士夫 訳  
日刊工業新聞社
- [4] Oh/FM別冊 「FILES Oh/FM」  
日本ソフトバンク
- [5] 「バイオリズムとはなにか」  
田多井吉之介  
講談社ブルーバックス
- [6] 「MSXビギナーズBASIC」  
児玉真之  
アスキー出版局

## MSX グラフィック・ワークブック

1984年5月5日 初版発行  
定価1,500円

著者 桜田幸嗣・筑島聡  
発行者 塚本慶一郎  
発行所 株式会社アスキー

©1984 ASCII Corporation. Printed in Japan.

本書は著作権法上の保護を受けています。本書の一部あるいは全部について（ソフトウェア及びプログラムを含む）、株式会社アスキーから文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

編集担当 竹山正寿  
表紙担当 郷 啓子  
印刷 図書印刷株式会社

ISBN4-87148-730-X C0055 ¥1500E



# キャラクタコード表

コード	*キャラクタ	対応するキー	コード	コード	キャラクタ	コード	コード	キャラクタ	コード	コード	キャラクタ	コード	コード	キャラクタ	コード	コード	キャラクタ	コード	コード	キャラクタ	コード				
			16進	10進	16進	10進	16進	10進	16進	10進	16進	10進	16進	10進	16進	10進	16進	10進	16進	10進	16進	10進			
0	ヌル	—	&H00	32		&H20	64	@	&H40	96	`	&H60	128	♠	&H80	160		&HA0	192	タ	&HC0	224	た	&HE0	
1	↑ コントロール・キャラクタ ↓	月	CTRL A	&H01	33	/	&H21	65	A	&H41	97	a	&H61	129	♥	&H81	161	。	&HA1	193	チ	&HC1	225	ち	&HE1
2		火	CTRL B	&H02	34	°	&H22	66	B	&H42	98	b	&H62	130	♣	&H82	162	〒	&HA2	194	ツ	&HC2	226	つ	&HE2
3		水	CTRL C	&H03	35	#	&H23	67	C	&H43	99	c	&H63	131	♦	&H83	163	」	&HA3	195	テ	&HC3	227	て	&HE3
4		木	CTRL D	&H04	36	\$	&H24	68	D	&H44	100	d	&H64	132	○	&H84	164	,	&HA4	196	ト	&HC4	228	と	&HE4
5		金	CTRL E	&H05	37	%	&H25	69	E	&H45	101	e	&H65	133	●	&H85	165	・	&HA5	197	ナ	&HC5	229	な	&HE5
6		土	CTRL F	&H06	38	&	&H26	70	F	&H46	102	f	&H66	134	を	&H86	166	ヲ	&HA6	198	ニ	&HC6	230	に	&HE6
7		日	CTRL G	&H07	39	'	&H27	71	G	&H47	103	g	&H67	135	あ	&H87	167	ッ	&HA7	199	ヌ	&HC7	231	ぬ	&HE7
8		年	CTRL H	&H08	40	(	&H28	72	H	&H48	104	h	&H68	136	い	&H88	168	イ	&HA8	200	ネ	&HC8	232	ね	&HE8
9		円	CTRL I	&H09	41	)	&H29	73	I	&H49	105	i	&H69	137	う	&H89	169	ウ	&HA9	201	ノ	&HC9	233	の	&HE9
10		時	CTRL J	&H0A	42	*	&H2A	74	J	&H4A	106	j	&H6A	138	え	&H8A	170	エ	&HAA	202	ハ	&HCA	234	は	&HEA
11		分	CTRL K	&H0B	43	+	&H2B	75	K	&H4B	107	k	&H6B	139	お	&H8B	171	オ	&HAB	203	ヒ	&HCB	235	ひ	&HEB
12		秒	CTRL L	&H0C	44	,	&H2C	76	L	&H4C	108	l	&H6C	140	や	&H8C	172	ヤ	&HAC	204	フ	&HCC	236	ふ	&HEC
13		百	CTRL M	&H0D	45	-	&H2D	77	M	&H4D	109	m	&H6D	141	ゆ	&H8D	173	ユ	&HAD	205	ヘ	&HCD	237	へ	&HED
14		千	CTRL N	&H0E	46	.	&H2E	78	N	&H4E	110	n	&H6E	142	よ	&H8E	174	ヨ	&HAE	206	ホ	&HCE	238	ほ	&HEE
15		万	CTRL O	&H0F	47	/	&H2F	79	O	&H4F	111	o	&H6F	143	つ	&H8F	175	ッ	&HAF	207	マ	&HCF	239	ま	&HEF
16		π	CTRL P	&H10	48	0	&H30	80	P	&H50	112	p	&H70	144		&H90	176	-	&HB0	208	ミ	&HD0	240	み	&HFO
17		┌	CTRL Q	&H11	49	1	&H31	81	Q	&H51	113	q	&H71	145	あ	&H91	177	ア	&HB1	209	ム	&HD1	241	む	&HF1
18		└	CTRL R	&H12	50	2	&H32	82	R	&H52	114	r	&H72	146	い	&H92	178	イ	&HB2	210	メ	&HD2	242	め	&HF2
19		┌	CTRL S	&H13	51	3	&H33	83	S	&H53	115	s	&H73	147	う	&H93	179	ウ	&HB3	211	モ	&HD3	243	も	&HF3
20		└	CTRL T	&H14	52	4	&H34	84	T	&H54	116	t	&H74	148	え	&H94	180	エ	&HB4	212	ヤ	&HD4	244	や	&HF4
21		┌	CTRL U	&H15	53	5	&H35	85	U	&H55	117	u	&H75	149	お	&H95	181	オ	&HB5	213	ユ	&HD5	245	ゆ	&HF5
22		└	CTRL V	&H16	54	6	&H36	86	V	&H56	118	v	&H76	150	か	&H96	182	カ	&HB6	214	ヨ	&HD6	246	よ	&HF6
23		┌	CTRL W	&H17	55	7	&H37	87	W	&H57	119	w	&H77	151	き	&H97	183	キ	&HB7	215	ラ	&HD7	247	ら	&HF7
24		└	CTRL X	&H18	56	8	&H38	88	X	&H58	120	x	&H78	152	く	&H98	184	ク	&HB8	216	リ	&HD8	248	り	&HF8
25		┌	CTRL Y	&H19	57	9	&H39	89	Y	&H59	121	y	&H79	153	け	&H99	185	ケ	&HB9	217	ル	&HD9	249	る	&HF9
26		└	CTRL Z	&H1A	58	:	&H3A	90	Z	&H5A	122	z	&H7A	154	こ	&H9A	186	コ	&HBA	218	レ	&HDA	250	れ	&HFA
27		┌		&H1B	59	;	&H3B	91	[	&H5B	123	┌	&H7B	155	さ	&H9B	187	サ	&HBB	219	ロ	&HDB	251	ろ	&HFB
27		└	⊗	&H1C	60	<	&H3C	92	Y	&H5C	124	└	&H7C	156	し	&H9C	188	シ	&HBC	220	ワ	&HDC	252	わ	&HFC
29		大		&H1D	61	=	&H3D	93	]	&H5D	125	┌	&H7D	157	す	&H9D	189	ス	&HBD	221	ン	&HDD	253	ん	&HFD
30		中		&H1E	62	>	&H3E	94	^	&H5E	126	└	&H7E	158	せ	&H9E	190	セ	&HBE	222	°	&HDE	254		&HFE
31		小		&H1F	63	?	&H3F	95		&H5F	127		&H7F	159	そ	&H9F	191	ソ	&HBF	223	°	&HDF	255		&HFF

\* このグラフィックキャラクタをPRINT, LPRINT命令で表示する時には, CHR\$(1)+CHR\$(n) とする。

