

コンピュータ教育における 教育的インターフェイス

市ノ瀬 慎一*

Pedagogical Interfaces in the Computer-Programming Training

Shin-ichi ICHINOSE

§ 1. ま え が き

従来のコンピュータ教育において、プログラミング言語を学習させても、それだけでは論理性が育つとは限らなかった。それは、すべてをかりそめの約束ごととしてとらえ、暗記と手続きに埋没する中で、記号操作の技術に習れてしまうことが起こるからである。基本的に、論理というものは、自分自身に対して説得していくプロセスの記述である。私たちは、ものごとに関連を持たせ、まとまりをつくり、構造をつくり出そうとして生きているからこそ、そういった構造化のできないものを、自分自身に対して説得することができないのである。こうした人間の情報処理能力の弱さに思い至る時、「人間にとって、コンピュータとはいかなるものか？」と改めて問い直したくなる。コンピュータが私たちの心の中にあった、シンボルを操作するという本性を拡大した存在であることに気づくのは容易であろう。ここから、シンボル活動に伴なうプラスの面もマイナスの面も、必然的に拡大されていくという宿命が生じてくる。コンピュータ教育を考える時、第一に、この点を踏まえる必要があると考えられる。

文字使用から始まった人類のシンボル活動が、コンピュータの出現によって、その頂点に達し、同時に、シンボル使用の歪みも拡大の一途をたどってきた。その結果、機械的であることが、すなわち非人間的であるかの如きイメージを人びとの心に定着させてしまった。しかし、ここで考えなければならないことは、機械的であることが非人間的になるのもシンボル使用の歪みから起こるのであれば、機械的であることを人間的にするのも、それを正しいシンボル使用のコンテキストの中に位置づけることを通じて可能となることである。こうしたシンボル活動の抽象性と形式性をもたらすマイナス面を克服する方策もまた、私たち自身の心の中から生み出し、それをコンピュータ教育に実現させていくことが望まれよう。

今や、コンピュータは、人びとが「わかる」「わかりあう」ための、媒体として利用され始めた。コンピュータを用いて「絵」を描いたり、その「絵」を動かしたりできるようになるにつれて、シンボル本来の多角的な機能を取戻す方向が見えてきた。私たちは、今や、コンピュータを媒体として、自らに適したシンボルを生み出し、それを使いこなすことによって、ものごとをよりよく「わかる」ことができるようになった。コンピュータは

* 情報科学研究室（昭和63年9月30日受理）

確かに「考える」ことができる。しかし、私たちを離れて「勝手に」考えるのではなく、私たちと共に考える機能こそ、論理性を育てるコンピュータ教育が特に必要とするものである。

本稿では、こうした目的意識の下に、対話型グラフィックスのシステム化を通じて、学習者がコンピュータと共に考えていく知的インターフェイスを作成したい。

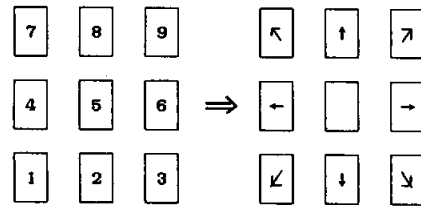
§ 2. モデル設計思想

コンピュータ教育において、人間と極く自然な相互作用ができるシステム作りを実施する場合、コンピュータが画面にメッセージを出し、人間がキーボードを通じて意志を伝えるというパターンが基本となる²⁾。このような思想の具体化は、コンピュータを用いて絵を描こうとする試みに始まると言える。そこで、「ペン」と「画用紙」に相当するものを、あらかじめ構成しておき、学習者は、この枠組みを出発点として、好みの機能を考案し、付加していくことによって、自らに適したシンボルを生み出すことができよう³⁾。このような知的空間を構築するための1つの素材として、MS-DOSとBASICの組み合わせを選んだ⁴⁾。

本節では、まず本モデル設計の出発点であるグラフィック・カーソルについて述べる。次に、学習者がコンピュータを媒体として、自らに適したシンボルを生み出していく舞台となる機能空間の構築について述べる。

2.1 グラフィック・カーソル

「ペン」の役割をするグラフィック・カーソル³⁾は、極めて単純な発想から作ることができる。Fig. 1に示すように、矢印がテンキー番号の方向に移動し、方向として4つしかないので、それぞれのキーを方向ごとに割りふればよい。つまり、リスト-1に示すように、1を入力すれば、左に行く時と下に行く命令の時だけ真を表すように設計しておく。ここで、130行のダミー変数XXとYYは前に書いた矢印を消すために、その座標を一時的に記憶しておく器であり、X_iのためにXX、YのためにYYを用意してある。その受け渡しは消し終わった時点で行うので、310行のコマンドが受け渡しになる。もちろん、ここで述べたグラフィック・カーソルはマウスに置き換えると、もっと簡単に絵が描ける。しかし、ここでは、プログラムの世界でペンの動きをイメージ化することに意義があるので、マウスの使用は前提としない。



テンキーによるグラフィック・カーソルの移動

Fig. 1

```

10          'リスト-1
20          '*****
30          '  First Setting  *
40          '*****
50  SCREEN 3
60  CONSOLE 0,25,0,1
70  DEFINT A-Z
80  DIM G(52)
90  LINE (0,0)-(15,15),5
100 LINE (0,0)-(0,7),5
110 LINE (0,0)-(7,0),5
120 GET@(0,0)-(15,15),G
130 X=320:Y=200:XX=X:YY=Y
140 C=6
150 CLS 3
160 POINT(X,Y)
170
180          '*****
190          '      Inkey      *
200          '*****
210 *MAIN
220 PUT@(X,Y),G,XOR
230 *INKC:'WHILE INKEY#<>':"WEND
240 *INKA:K#=INKEY#
250 IF K#="" THEN *INKA
260 IF INSTR("123",K#) THEN Y=Y+1
270 IF INSTR("789",K#) THEN Y=Y-1
280 IF INSTR("147",K#) THEN X=X-1
290 IF INSTR("369",K#) THEN X=X+1
300 ******
310 PUT@(XX,YY),G,XOR
320 XX=X:YY=Y
330 LINE-(X,Y),C
340 GOTO *MAIN

```

```

10          'リスト-2
20          '*****
30          '   First Setting   *
40          '*****
50 SCREEN 3
60 CONSOLE 0,25,0,1
70 DEFINT A-Z
80 DIM G(52)
90 LINE (0,0)-(15,15),5
100 LINE (0,0)-(0,7),5
110 LINE (0,0)-(7,0),5
120 GET@(0,0)-(15,15),G
130 X=320:Y=200:XX=X:YY=Y
140 C=6
150 CLS 3
160 POINT(X,Y)
170          '*****
180          '   Inkey   *
190          '*****
200 *MAIN
210 PUT@(X,Y),G,XOR
220 *INKC:'WHILE INKEY*<>'":WEND
230 *INKA:K$=INKEY$
240 IF K$="" THEN *INKA
250 IF INSTR("123",K$) THEN Y=Y+1
260 IF INSTR("789",K$) THEN Y=Y-1
270 IF INSTR("147",K$) THEN X=X-1
280 IF INSTR("369",K$) THEN X=X+1
290 IF INSTR("=-+*/-SLQ",K$) THEN GOSUB *SUBCALL
300 PUT@(XX,YY),G,XOR
310 XX=X:YY=Y
320 LINE-(X,Y),C
330 GOTO *MAIN
340          '*****
350          '*   Function Call   *
360          '*****
370 *SUBCALL
380 SK=INSTR("=-+*/-SLQ",K$)
390 ON SK GOSUB *FUNC,*FUNC,*FUNC,*FUNC,*FUNC,*FUNC,*FUNC,*FUNC
400 RETURN
410          '*****
420          '*   Function   *
430          '*****
440 *FUNC
450          '.....
460          '.....
470          '.....
480 RETURN

```

2.2 機能空間の構築

コンピュータを媒体として、多元的なシンボルを生み出していく舞台は、学習者の興味の進展につれ、付加機能の変更を余儀なくされることも出てくる。それゆえ、機能空間の変更・拡張は容易でなければならない。リスト-2に示すように、このモデル設計では、学習者が容易に機能の変更・拡張ができるように、INSTR文とON~GOSUB~文を組み合わせ、その中の変数を選択することで実現されるよう設計してある。リスト-2では、空の機能空間だけを用意し、このプログラムを実行した場合には、とりあえず「*SUBCALL」に飛び、そこで今のところ全て「*FUNC」に飛ぶようにしてある。このスペースこそ、学習者のコンピュータとの関わりの強さに従って、豊かな内容を持ち、自分自身の可能性を引き出し、それを伸ばしていける部分である。

これが、コンピュータ教育において、学習者とコンピュータの相互交流を支援する立場に立ったアプローチである。次節では、そのための具体的なシステム構成について述べてい。

§ 3 . システム構成

前節の枠組に諸機能を付加して、知的空間を拡大させていく具体例について説明する。

3.1 ソフトウェア構成

ソフトウェア構成を Fig. 2 に示す。各ソフトウェアの概要は次のとおりである。

① 初期設定

グラフィック・カーソルの矢印を16×16ドットの大きさで描く。大きさ637×339の画用紙を作成する。

② メインルーチン

線を描くルーチンを定義する。グラフィック・カーソルが画用紙枠をはみ出した時の処置および画面消去の手続を述べる。

③ 機能選択

付加すべき諸機能の割りつけを定義する。

3.2 機能付加

コンピュータ教育において、新しいアルゴリズムを見つけ出すことは、かなり高度な知的活動である。プログラム言語そのものの学習を主とするのでない限り、むしろ、アルゴリズムの枠組を与えて、学習者の具体的な興味に即して、ツール・プログラムに何かを付け加えることから始めるのが適当と考えられる。コンピュータと対話させながら、知的空間の拡大をめざす、本モデルでは、以下に述

べる8つの機能を学習者のレベルに応じて、プログラム化させていく方法を採用しよう。8つの機能を Fig. 3に示すように、その機能名を命名しておく。全体のシステムプログラムはリスト-3に示されている。

① 色を変えて描いてみたい (*CLR)

* インタラクティブ・グラフィックス *

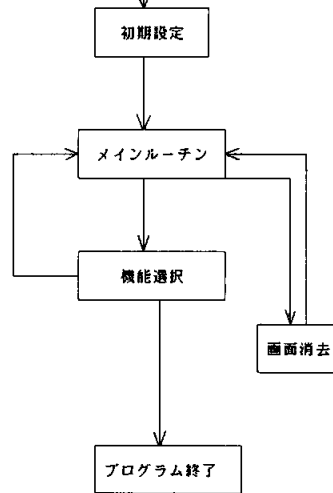


Fig. 2

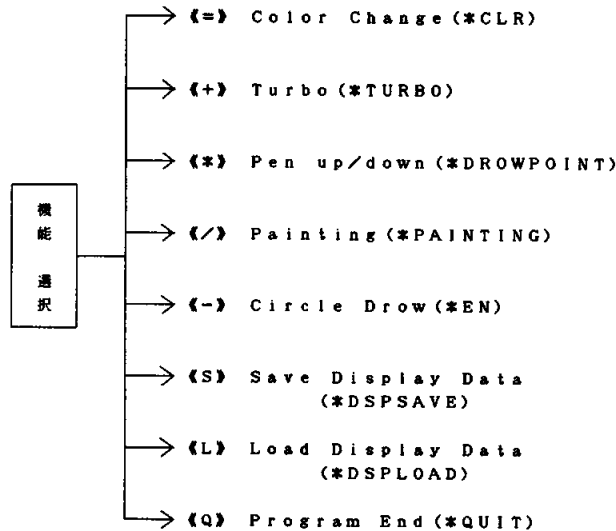


Fig. 3

色を変えるには、LINE文のカラーモードを決める変数「C」の値を変えるサブルーチンを作ればよい。リスト-2のプログラムでFunctionとして用意しておいた「=」を、カラーチェンジモードとする。580行の色変更のサブルーチン呼び出す度にカラーナンバーを1つつアップするように設計され、また、カラーナンバーが7以上になるようなら0に戻すように定義されている。

- ② 1ドットずつ動くばかりではなく、10ドットずつでも動くようにしたい（*TURBO）

TURBO命令も「+」で呼び出す毎に±10動くか、±1かを切り替えている。そのための切替変数が「T」で、メインルーチンも変更されている。今まで、単純にXY軸方向に±1としていたものを、Tとおいた。初期設定で変数Tを10と設定してあるのは、一度、「TURBO OFF」のメッセージを出しておきたいため、最初にメッセージをおくようにすれば、このような無理をする必要はない。

- ③ いつも描くばかりでなく、描かずに移動させたい（*DROWPOINT）

ペンのアップ・ダウンもTURBO命令と同じ発想である。特に、解析するまでもないが、リスト-3の490行のLINE文が変更されている点に留意されたい。

- ④ 色塗りをしたい（*PAINTING）

850行から始まる色塗り命令は、Fig 4に示すように、塗る位置、塗る色、枠の色を順次、サブルーチン（*GCURSOR）、サブルーチン（*PAINTCOLOR）を呼び出ししながら、実行していく。Fig. 6および Fig 7に示す二つのサブルーチン

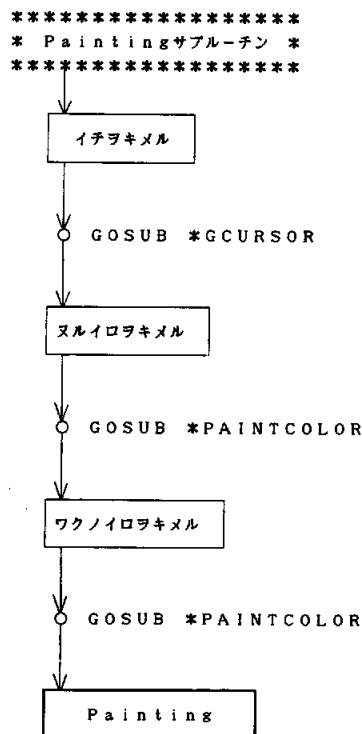


Fig. 4

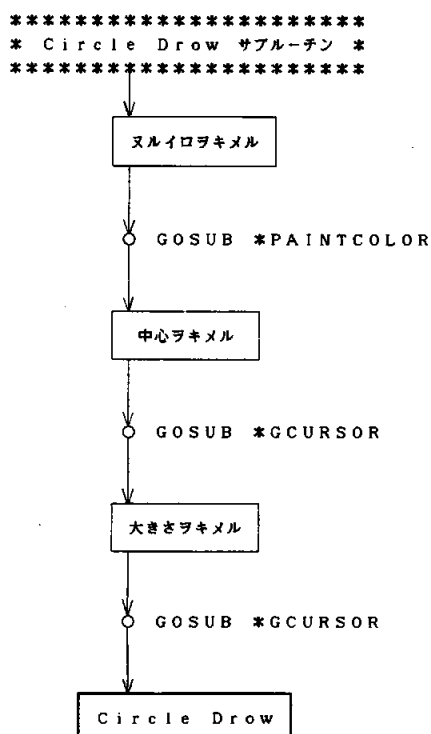


Fig. 5

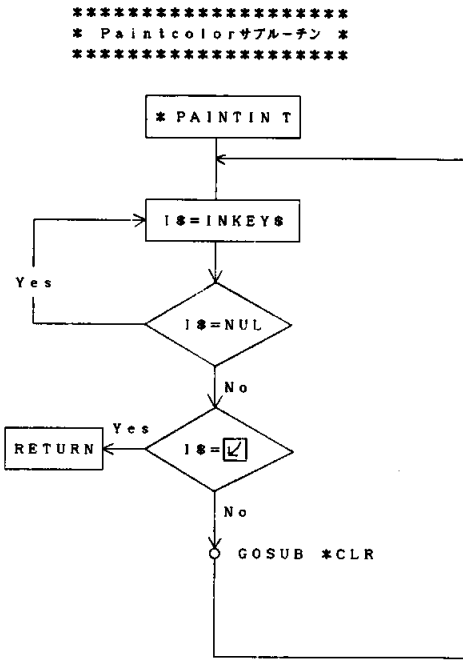


Fig. 6

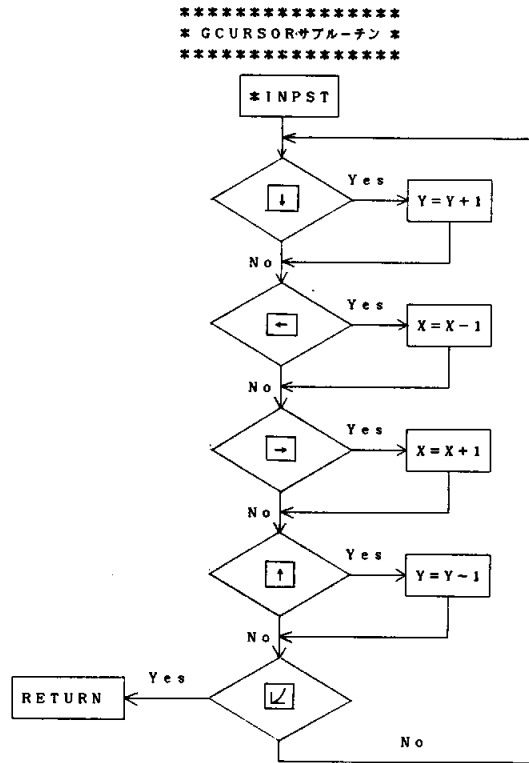


FIG. 7

は、次の円を描く場合にも呼び出され、本モデルで基本的な役割を演ずる。

⑤ 円を描いてみたい (*EN)

1190行の円命令も、Fig. 5に示すように、塗る色、中心位置、大きさ(半径)の順番で、Fig. 6と Fig. 7に示したサブルーチン呼び出しつつ描くようにしてある。

⑥ 描いた絵をファイルに保存したい (*DSPSAVE)

⑦ 描いた絵をファイルからロードしたい (*DSPLOAD)

保存用のサブルーチンが1700行から始まる *DSPSAVEおよび *DSPLOADである。 *DSPSAVEを実行すると、B, R, G (青, 赤, 緑)の文字がファイル名の後に追加されて、それぞれが独立した3個のファイルとなる。SAVEとLOADの実行はファイル名の入力で行える。

⑧ 描いた絵を消して終了したい (*QUIT)

画面消法のサブルーチンが *CLEARDSPであり、プログラムを終了したい時は、サブルーチン (*QUIT)を実行する。

機能空間として、上述の8つを選んだが、学習者の興味の対象が拡大していくにつれ、学習者自身が工夫して、知的空間を豊かにしていくことが教育上好ましい。そうした目的に対応できるように、システムの枠組を単純な形に作り、機能を順次付加していきけるように設計した。こうした Training を通じて、学習者が何よりも、知的インターフェイスの開発の一端を、プログラム言語の世界に立って体験できると考えられる。


```

950 PRINT "[PAINT] スル イロヲキメテクタサイ-----";
960 GOSUB *PAINTCOLOR
970 C1=C
980 COLOR 7:LOCATE 0,24
990 PRINT "[PAINT] ツクノイロヲキメテクタサイ-----";
1000 GOSUB *PAINTCOLOR
1010 C2=C
1020 PAINT(X,Y),C1,C2
1030 PUT@(XX,YY),G,XOR
1040 C=C0:DRW=-1:GOSUB *DROWPOINT
1050 COLOR 7:LOCATE 0,24
1060 PRINT " ";
1070 RETURN
1080 '*****
1090 '* Paintcolor *
1100 '*****
1110 *PAINTCOLOR
1120 IF INKEYS<>"" THEN *PAINTCOLOR
1130 *PAINTINT
1140 IS=INKEYS
1150 IF IS="" THEN *PAINTINT
1160 IF IS=CHR$(13) THEN RETURN
1170 GOSUB *CLR
1180 GOTO *PAINTINT
1190 RETURN
1200 '*****
1210 '* Circle Draw *
1220 '*****
1230 *EN
1240 FOR LOOP=1 TO 500:NEXT LOOP
1250 COLOR 7:LOCATE 0,24
1260 PRINT "イロヲキメテクタサイ ";
1270 C0=C
1280 GOSUB *PAINTCOLOR
1290 FOR LOOP=1 TO 500:NEXT LOOP
1300 COLOR 7:LOCATE 0,24
1310 PRINT "チュウシンヲキメテクタサイ ";
1320 GOSUB *GCURSOR
1330 X0=X:Y0=Y
1340 FOR LOOP=1 TO 500:NEXT LOOP
1350 COLOR 7:LOCATE 0,24
1360 PRINT "オキキヲキメテクタサイ ";
1370 GOSUB *GCURSOR
1380 PUT@(X0,Y0),G
1390 RC=SQR((X-X0)^2+(Y-Y0)^2)
1400 IF RC<1 THEN GOTO *ESCCIRCLE
1410 '
1420 CIRCLE(X0,Y0),RC,C
1430 C=C0-1:GOSUB *CLR:DRW=-1:GOSUB *DROWPOINT
1440 LOCATE 0,24
1450 PRINT SPC(60);
1460 PUT@(X,Y),G
1470 RETURN
1480 *ESCCIRCLE
1490 PUT@(X0,Y0),G
1500 X=X0:Y=Y0:XX=X:YY=Y
1510 LOCATE 0,24:PRINT SPC(60,)
1520 RETURN
1530 '*****
1540 '* Gcursion *
1550 '*****
1560 *GCURSOR
1570 PUT@(X,Y),G,XOR
1580 *INPST
1590 PUT@(X,Y),G,XOR
1600 *XYCHECK
1610 IF INP($HEA)=&HFD THEN Y=Y+1 'シタ
1620 IF INP($HEA)=&HFB THEN X=X-1 'ヒタリ
1630 IF INP($HEB)=&HFB THEN X=X+1 'ミキ
1640 IF INP($HEB)=&HFD THEN Y=Y-1 'ウエ
1650 PUT@(XX,YY),G,XOR
1660 XX=X:YY=Y
1670 IF INP($HE1)=&H7F THEN *GEND 'return key
1680 GOTO *INPST
1690 *GEND
1700 RETURN
1710 '*****
1720 '* Save Display Data *
1730 '*****
1740 *DPSAVE
1750 WHILE INKEYS<>"":WEND
1760 CONSOLE 23,1
1770 COLOR 7:LOCATE 0,23
1780 INPUT "キョウシキヲフイルノナマエヲイレテクタサイ: ",FF$
1790 CONSOLE 0,24
1800 IF INSTR(FF$,".") THEN RETURN
1810 PUT@(X,Y),G
1820 DEF SEG=&HA800:BSAVE FF$+".blu",0,&H7CFF
1830 DEF SEG=&HB000:BSAVE FF$+".red",0,&H7CFF
1840 DEF SEG=&HB800:BSAVE FF$+".grn",0,&H7CFF
1850 PAINT(0,0),0,7
1860 PUT@(X,Y),G
1870 RETURN
1880 '*****
1890 '* Load Display Data *
1900 '*****
1910 *DSPLOAD
1920 WHILE INKEYS<>"":WEND
1930 CONSOLE 23,1
1940 COLOR 7:LOCATE 0,23
1950 INPUT "キョウシキヲフイルノナマエヲイレテクタサイ: ",FF$
1960 CONSOLE 0,24
1970 IF INSTR(FF$,".") THEN RETURN
1980 PUT@(X,Y),G
1990 DEF SEG=SEGPTH(2)
2000 DEF SEG=&HA800:BLOAD FF$+".blu"
2010 DEF SEG=&HB000:BLOAD FF$+".red"
2020 DEF SEG=&HB800:BLOAD FF$+".grn"
2030 PUT@(X,Y),G
2040 C=C-1:GOSUB *CLR
2050 T=10:GOSUB *TURBO
2060 DRW=-1:GOSUB *DROWPOINT
2070 RETURN
2080 '*****
2090 '* Display Clear *
2100 '*****
2110 *CLEARDSP
2120 CONSOLE 23,1
2130 COLOR 7:LOCATE 0,23
2140 PRINT "カメシキヲケシマス。イテスカ(y/n)";
2150 YNS=INPUT$(1)
2160 CONSOLE 0,24
2170 IF YNS<>"y" AND YNS<>"Y" THEN RETURN
2180 CLS 3
2190 X=320:Y=200:XX=X:YY=Y
2200 LINE(XMIN-1,YMIN-1)-(XMAX+1,YMAX+1),7,B
2210 PUT@(X,Y),G
2220 C=C-6:GOSUB *CLR
2230 T=10:GOSUB *TURBO
2240 DRW=-1:GOSUB *DROWPOINT
2250 RETURN
2260 '*****
2270 '* Program End *
2280 '*****
2290 *QUIT
2300 WHILE INKEYS<>"":WEND
2310 CLS 3
2320 END

```


§ 4. む す び

本稿では、教育的インターフェイスの立場から、学習者がコンピュータと共に考えるためのシステム作りについて述べた。本モデルの特徴をまとめると、以下のようになる。

- (1) コンピュータを媒体として、学習者自身が自らに適したシンボルを生み出していくための方向性とその枠組を実現している。
- (2) 学習者が知的インターフェイスの開発の一端を、プログラム言語のレベルを通して体験できる。
- (3) 学習者の興味の対象とレベルに応じて、種々の機能付加およびそのバリエーションが容易に行える。

今後は、より多元的な機能をもったシンボルの作り出せるような教材開発を図っていきたい。

参 考 文 献

- 1) 佐伯 胖「コンピュータと教育」、岩波書店、東京(1986)。
- 2) 河西朝雄「構造化BASIC」、技術評論社、東京(1985)。
- 3) 知能システム研究会「BASICでつくる脳の情報システム」、啓学出版、東京(1987)。
- 4) 黒田康太「基礎MS-DOS&BASIC」、東京電気大学出版局、東京(1988)。