

Macintosh のグラフィック機能の研究

～ HyperTint をモデルにして ～

小西 忠男・坂本 優子

Study of Graphic Function of Macintosh

—— How to Use HyperTint ——

Tadao KONISHI and Yuko SAKAMOTO

1. はじめに

小学校でコンピュータが導入されることが決まり、教育の方法や技術についての学校の動向も、現在、しだいに変わりつつある。今日では、教材の作成のように、教師が主体的に関与できるものが重要になってきている。また、いろいろなメディアを教師がうまく組み合わせて使うようにもなっている。教師が教育のための道具として用いるほかに、児童が自らの学習の道具として使う状況も見られるようになってきた。

コンピュータは、さまざまな用途に使われ、授業の改善に貢献するようになった。ビデオディスクと連動したり、イメージスキャナから画像を取り込んだりさまざまな工夫もなされている。

コンピュータに関するさまざまな影と批判のあるなかで、情報化社会における教育の方法と技術は、授業の設計、実践、計画の改善、教育メディアの有効利用などを中心にして、新しい方向に展開しつつある。

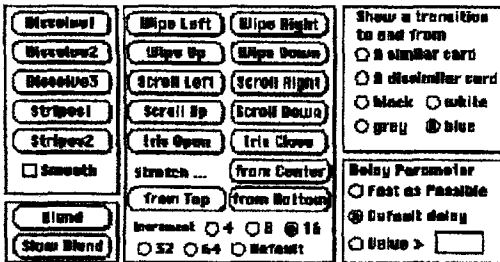
そこで、教材作成ツールとして有意義である HyperCard について研究を始めた。HyperCard は、使い方によって、ワープロになったり、

データベースになったり、表計算ソフトやグラフィックソフトにもなる。しかし、HyperCard は、カラーが使いにくいので、教材内容の提示にとって有効なディスプレイ画面を作成するのに支障がある。そんな中、最近になって、この点が解決された HyperTint というソフトが開発された。しかしながら、このソフトの使い方についてマニュアルが不完全なので、HyperTint のデモンストレーションプログラムを解析することにより、マニュアルづくりを試みた。まだ、解析できた範囲でカラー命令の事例をいくつか作成してみた。

2. HyperTint について

HyperTint は、HyperCard 上でカラーを使うことが可能である。カード、バックグラウンドの後に 3 枚目のカードがある。その 3 枚目のカード上で、色をつけたり、色を変えたりすることができるようになっている。3 枚目のカードを操作するには、5 つのオブジェクト（スタック、バックグラウンド、カード、ボタン、フィールド）のスクリプトの中に、C 言語を使って作成した XCMD'S（拡張コマンド）、XFNC（拡張ファンクション）を使って HyperTalk を書いていく。では、例として HTVisual

について説明する。HTVisual は、HyperCard
では、effect visual (視覚効果) にあたる。つまり、これは、現在使っているカードから他のカードへ移動させる方法である。教材としてHyperCardを使うとき、どのような方法で他のカードに移動するのが良い方法かを選べる。HTVisual には、次のような視覚効果がある。



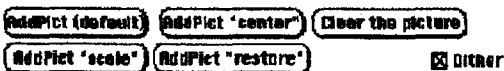
また、色を指定したり、速さの調節もできる。スクリプトには、次のような形式で書く。

HTVisual "effect","color","rectangle","delay",
"increment"

この中で、省略するものは、"" とする。例えば、

HTVisual "Scroll Left","65535,0,0","", "", "1"
と書く。

次に HTAddPict について説明する。HTAddPict は、次のようなことができる。



HTAddPict は、カードに色をつける上で、たいへん重要な役割をもっている。カード上で、カラーを使うには、次の方法がある。

○Region で色をつける。

HyperTint では、10個の Region を指定できる。また、ウインドウ "HT Color Panel" を使って、色を決める方法もある。この中で、クリックするとバックグラウンドカラーを、シフトキーを押してクリックするとフォアグラウン

ドカラーを変えることができる。また、オプションキーを押してクリックするとバックグラウンドを選択して変えることができる。

○ドットで色をつける。

拡張コマンド HTAddPict を使って、色をつける。

3. 自作描画ツールについて

いろいろな拡張コマンドを使ってカラーで簡単な絵が描けるカードを作ってみた。このカードでは、すべてボタンスクリプトで操作するようにしてみた。では、どのようなことができるようになったのか説明する。

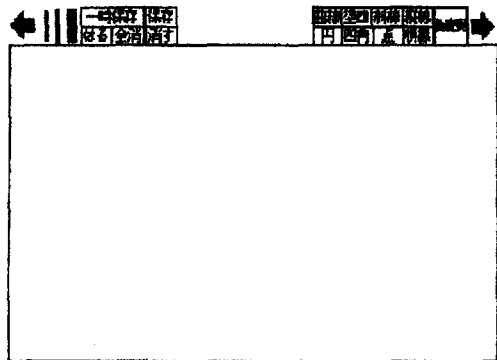


図 1

図 1 は、実際に HyperTint 上でカラーの絵を描くことができるように設計されたカードである。このカードの大きな四角は実際に絵を描くスペースであり、同時に、いくつかの小さな四角などで作成されているボタンの内容を受けて、機能するように設計されたボタンになっている。このボタンスクリプトでは、mousestill-down のハンドラでさまざまな描画機能を分岐させている。

```
on mousestilldown
  global iroa,sena,zukei
  if zukei=1 then tatesen
```

```

if zukei=3 then sikakunuru
if zukei=4 then kesu1
if zukei=5 then yokosen
if zukei=8 then kesu2
if zukei=9 then kyokus
if zukei=10 then sikaku
end mousestildown

```

それぞれの小さな四角などで作成されているボタンは、さまざまな描画機能を受け持っているが、その内容を順に説明する。

1) 色変更ボタンで色を決める。

*ここでは、HyperTint デモンストレーションの色を決めるパレット "Pick a new color" を HTUPickColor で呼び出し、その結果を it で受けて、取り入れた。

——色を決めるハンドラ——

```

on mouseUp
  global iroa,sena
  HTUPickColor "Pick a new color", the short
  name of me
  if the result is not empty then
    if the result is not "Cancel" then answer the
    result
  else
    set the name of me to it
    put it into iroa
    HTBackground the short name of me
    HTAddPict "none",the rect of button id 1
  end if
end mouseUp

```

2) 線の太さを決める。

*色を決めたあとで、線の太さを決めるボタンをクリックする。

——線の太さを決めるハンドラ——

```

on senfutura
  global iroa,sena,futura
  put mouseh into mx

```

```

  put mousev into my
  if sena = 1 then put "mh,mv,mh + 3,mv + 3"
  into sena
  if sena = 2 then put "mh,mv,mh + 6,mv + 6"
  into sena
  if sena = 3 then put "mh,mv,mh + 9,mv + 9"
  into sena
end senfutura

```

3) 点を描く。

*マウスを押すとその位置に点を描く。

——ボタン点のハンドラ——

```

on mousedown
  global iroa,sena,zukei
  if zukei=7 then
    put the mouseh into h
    put the mousev into v
    htbackground iroa
    htaddpict "none","h,v,h+sena,v+sena"
  end if
end mousedown

```

4) 面塗長方形を描く。

*開始点を決め、マウスを押したままの状態では長方形の大きさを決め、決まったらマウスをアップする。

——ボタン塗四のハンドラ——

```

on sikakunuru
  global iroa,sena,zukei
  put the mouseh into ax
  put the mousev into ay
  repeat until the mouse is up
    put the mouseh into h
    put the mousev into v
    htbackground iroa
    htaddpict "none","ax,ay,h,v"
  end repeat
end sikakunuru

```

5) 長方形を描く。

*開始点を決め、マウスを押したままの状態を終了点までドラッグし、マウスをアップする。

——ボタン四角のハンドラ——

```
on sikaku global iroa,sena,zukei,futosa
  put the mouseh into ax
  put the mousev into ay
  repeat until the mouse is up
    put the mousev into v
    put the mouseh into h
    htbackground iroa
    htaddpict "none","ax,v,ax+sena,v+sena"
    htbackground iroa
    htaddpict "none","h,ay,h+sena,ay+sena"
  end repeat
  repeat with i=0 to h-ax
    htbackground iroa
    htaddpict "none","ax+i,v,ax+i+sena,v+sena"
    htaddpict "none","ax+i,ay,ax+i+sena,ay+sena"
  end repeat
  repeat with i=0 to v-ay
    htbackground iroa
    htaddpict "none","h,ay+i,h+sena,ay+i+sena"
    htaddpict "none","ax,ay+i,ax+sena,ay+i+sena"
  end repeat
end sikaku
```

6) 縦線を描く。

* 開始点を決め、マウスを押したままの状態
縦線の長さを決め、決まったらマウスをアップ
する。縦線を描くときは、ゆっくりドラッグす
る。

——ボタン縦線のハンドラ——

```
on tatesen
  global iroa,sena,zukei,futosa
  put the mouseh into ax
  put the mousev into ay
  repeat until the mouse is up
    put the mousev into v
    htbackground iroa
    htaddpict "none","ax,v,ax+sena,v+sena"
```

end repeat

end tatesen

7) 横線を描く。

* 開始点を決め、マウスを押したままの状態
横線の長さを決め、決まったらマウスをアップ
する。横線を描くときは、ゆっくりドラッグす
る。

——ボタン横線のハンドラ——

```
on yokosen
  global iroa,sena,zukei,futosa
  put the mouseh into ax
  put the mousev into ay
  repeat until the mouse is up
    put the mouseh into h
    htbackground iroa
    htaddpict "none","h,ay,h+sena,ay+sena"
  end repeat
end yokosen
```

8) 曲線を描く。

* 開始点を決め、マウスを押したままの状態
曲線を描く。曲線を描くときは、ゆっくりド
ラッグする。

——ボタン曲線のハンドラ——

```
on kyokusen
  global iroa,sena,zukei
  repeat until the mouse is up
    put the mouseh into h
    put the mousev into v
    htbackground iroa
    htaddpict "none","h,v,h+sena,v+sena"
  end repeat
end kyokusen
```

9) 斜線を描く。

* マウスをクリックして開始点を決め、もう一
度マウスをクリックして終了点を決める。

——ボタン斜線のハンドラ——

```
on naname
  global iroa,sena,zukei,ka,gx,gy
```

```

put ka+1 into ka
put ka mod 2 into j
if j=1 then
    put the mouseh into gx
    put the mousev into gy
else
    nanamekaku
end if
end naname

on nanamekaku
    global iroa,sena,zukei,ka,gx,gy
    put the mouseh into hx
    put the mousev into hv
    if (hx-gx)>(hv-gy) then
        put hx-gx into sa
        put (hv-gy)/sa into jy
        repeat with i=0 to sa
            put jy*i div 1 into jx
            htbackground iroa
            htaddpict "none", "gx+i,gy+jx,gx+i+
sena,gy+jx+sena"
        end repeat
    else
        put hv-gy into sa
        put (hx-gx)/sa into jy
        repeat with i=0 to sa
            put jy*i div 1 into jx
            htbackground iroa
            htaddpict "none", "gx+jx,gy+i,gx+jx+
sena,gy+i+sena"
        end repeat
    end if
    put 0 into ka
end nanamekaku

```

10) 円を描く。

* マウスをクリックして円の中心を決め、もう一度マウスをクリックすると中心との距離で半径が決まり、円を描く。

—— ボタン円のハンドラ ——

```

on en
    global iroa,sena,zukei,ka,gx,gy
    put ka+1 into

```

```

ka
put ka mod 2 into j
if j=1 then
    put the mouseh into gx
    put the mousev into gy
else
    enkaku
end if
end en

on enkaku
    global iroa,sena,zukei,ka,gx,gy
    put the mouseh+1 into h
    put the mousev+1 into v
    put h-gx into hkk
    repeat with i=1 to 160
        put sqrt(((hkk*sin(i*pi/80))^2+(hkk*cos
(i*pi/80))^2) div 1) into j1
        if i>40 then
            if i<120 then
                put 3.14+atan(sin(i*pi/80)/cos
(i*pi/80)) into j2
            else
                put atan(sin(i*pi/80)/cos(i*pi/80))
into j2
            end if
        else
            put atan(sin(i*pi/80)/cos(i*pi/80)) into
j2
        end if
        put gx+(j1*cos(j2)+0.5) div 1 into jx
        put gy+(j1*sin(j2)+0.5) div 1 into jy
        htbackground iroa
        htaddpict "none", "jx,jy,jx+sena,jy+sena"
    end repeat
end enkaku

```

11) 描いた絵を消す。

* マウスを押したままの状態で移動すると、部分的に消すことができる。

—— ボタン消すのハンドラ ——

```

on kesul
    global iroa,sena,zukei

```

```
repeat until the mouse is up
  set the cursor to cross
  put the mouseh into h
  put the mousev into v
  htbackground "white"
  htaddpict "none", "h,v,h+10,v+10"
end repeat
end kesu1
```

12) 描いた絵を全部消す。

* 消したい部分の開始点を決め、マウスを押したままの状態で移動すると、消すことができる。

——ボタン全消のハンドラ——

```
on kesu2
  global iroa,sena,zukei
  put the mouseh into ax
  put the mousev into ay
  repeat until the mouse is up
    put the mouseh into h
    put the mousev into v
    htbackground "white"
    htaddpict "none", "ax,ay,h,v"
  end repeat
end kesu2
```

13) 描いた絵を保存する。

* ここで保存すると他のアプリケーション (Super Paint 3.0 など) に作成したものを移動させることができる。

——ボタン一時保存のハンドラ——

```
on mouseUp
  htsavepict "0,30,448,280","clipboard"
end mouseUp
```

14) 描いた絵を別のカードに保存する。

* ここで保存すると別のカードに作成したものを移動させることができる。別のカードでは、修正できない。

——ボタン保存のハンドラ——

```
on mouseUp
```

```
  global iroa
  htsavepict "0,32,448,280","clipboard"
  go card id 20861
  htlock off
  htaddpict "pict","0,32,448,280","clipboard"
  set the showPict of this background to true
  go card id 24169
end mouseUp
```

——移動したカードのカードスクリプト——

```
on openCard
  clearallregions
  global iroa,sena
  send openCard to this background
  htaddpict "pict","0,32,448,280","clipboard"
  displayColorBoxes
end openCard

on displayColorBoxes
  restore it later
end displayColorBoxes

on lockfields which
  if which is empty then
    put true into which
  end if
  repeat with i=1 to the number of card fields
    set the locktext of card field i to true
  end repeat
end lockfields

on cr
  clearAllRegions
  HTWriteScheme the short name of this card
  if the result is not empty then
    answer "Error saving color scheme: " & the result
  end if
  go this card
end cr

on closecard
  clearallregions
  htsavepict "0,32,448,280","this card"
end closecard
```

15) 描いた絵を他のカード・アプリケーションにペーストする。

* ここでは、他のアプリケーション (Super Paint 3.0 など) で作成したものをそこで保存し、今まで難しかったカラーで取り込むということを HyperTint を用いることによって可能にした。こうして、HyperCard に取り込んで、カード上で、ボタン、フィールドと組み合わせ、簡単な修正ができるようになった。

——ボタンはるのハンドラ——

```
on mouseUp
  global iroa
  htaddpict "", "0,28,448,280", "clipboard"
end mouseUp
```

4. おわりに

この研究を通して、HyperCard を操作するた

めの言語である HyperTalk は、Pascal や C 言語などで作成されたさまざまな拡張機能をリソースとして HyperCard のリソースフォークに取り入れることにより、より多様な機能を備えることができるということが分かった。HyperTint のカラーグラフィック機能もこの拡張性を利用したものであって、今後さらに、有用な XCMD'S が開発されることが期待される。教育活動のさまざまな場面に導入され始めているコンピュータは、このようなソフトウェアがさらに強化されることによって、教育メディアとしての役割を真に果たすことが可能になるであらう。