情報処理 2 第 8 回

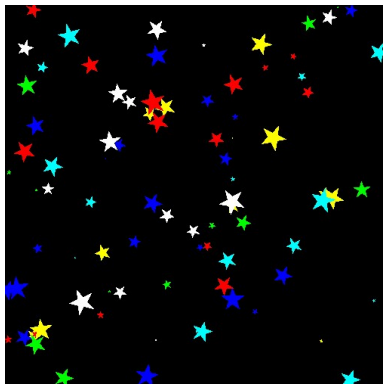
十進BASIC (6) グラフィックス入門

桂田 祐史 (かつらだ まさし)

2006 年 6 月 14 日

ホームページは http://www.math.meiji.ac.jp/~mk/syori2-2006/

前回の課題 5B (1)「正五角形の対角線で作られる星形を金色に塗る」どうなったでしょうか。私自身は二つの五角形が出て来ることに注目して、相似比を計算することで星形の頂点の座標を計算して図を描くことが出来ました (まだ銀色ですけど)。



ところで、課題 5B (2)「Koch 曲線を描く」を考えていて気付いたのですが、タートルグラフィックス (すぐ後で説明します) を使えば、この星形は簡単に描けることに気が付きました (そうして遊んでみたのが右側にある図です)。

そこでこちらの最初の予定からは少々脱線しますが、タートルグラフィクスの説明をします。

1 課題 5B への一つのヒント: タートルグラフィックス

1.1 LOGO のタートルグラフィックスとは

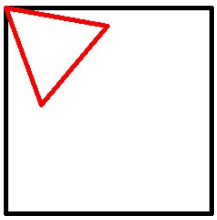
一時期、プログラミングの入門用の言語として、LOGO というものが注目されたことがあります。その特徴であるタートル・グラフィックスを利用すれば、まだ座標系の概念を学んでいないような小学生にも簡単に図形を描くプログラムが作れるということで「ちょっと面白い」話でした。

のタートル・グラフィックスとは、キャンバスの中を亀 (タートル) が歩く軌跡として図形を描画するというものです。亀に伝えられる指令として、「進む (指定された歩数だけ)」,「戻

る (指定された歩数だけ)」,「右に曲がる (指定された角度だけ)」,「左に曲がる (指定された角度だけ)」,「ペンを下ろす」,「ペンをあげる」などがあります。

例えば、「50 歩進み、90° 右に曲がる」を 4 回繰り返すことで 1 辺の長さが 50 歩の正方形を描くことができます。

```
REM TURTLE.BAS --- 亀よ正方形と正三角形を描いておくれ
OPTION ANGLE DEGREES
SUB right(t)
   LET direction=direction-t
END SUB
SUB left(t)
   LET direction=direction+t
END SUB
SUB walk(s)
   PLOT LINES: xp,yp;
   LET  xp=xp+s*COS(direction)
   LET  yp=yp+s*SIN(direction)
   PLOT LINES: xp,yp
END SUB
SUB jump(s)
   LET  xp=xp+s*COS(direction)
   LET  yp=yp+s*COS(direction)
   PLOT LINES: xp,yp
END SUB
REM ----------- start ------------
LET  L=100
SET WINDOW -1.1*L,1.1*L,-1.1*L,1.1*L
REM 亀の現在位置 (xp,yp), 方向 direction (x 軸となす角)
LET  direction=0
LET  xp=0
LET  yp=0
FOR i=1 TO 4
   CALL walk(L)
   CALL right(90)
NEXT i
CALL right(10)
SET LINE COLOR "red"
FOR i=1 TO 3
   CALL walk(L/2)
   CALL right(120)
NEXT i
END
```



1.2　試してみよう＆課題 6A

“TURTLE.BAS” をたたき台にすれば、以下は比較的簡単でしょう。

- 3 以上の自然数 n を与えられたとき、正 n 角形を描くプログラムを作れ。
- 正五角形の対角線で出来る星形を描くプログラムを作れ (塗りつぶしでなく線画ならば簡単です)。

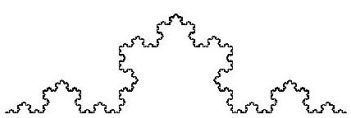
どちらかのプログラムを今日の課題 6A として提出して下さい。
このアイディアを生かせば、星形を塗り潰すプログラムもあまり難しく書けると思います。

1.3　参考: Koch 曲線を描くプログラム

課題とすることには難しさがあるかと思ったので、こちらがプログラム例を示すことにしました。

「$n/3$ 進み、左に 60° 曲がり、$n/3$ 進み、右に 120° 曲がり、$n/3$ 進み、左に 60° 曲がり、$n/3$ 進む」と、

このような図になりますが、「$n/3$ 進む」ところを「端点間の距離が $n/3$ の Koch 曲線を描く」に置き換えると、「端点間の距離が n の Koch 曲線を描く」ことができます。

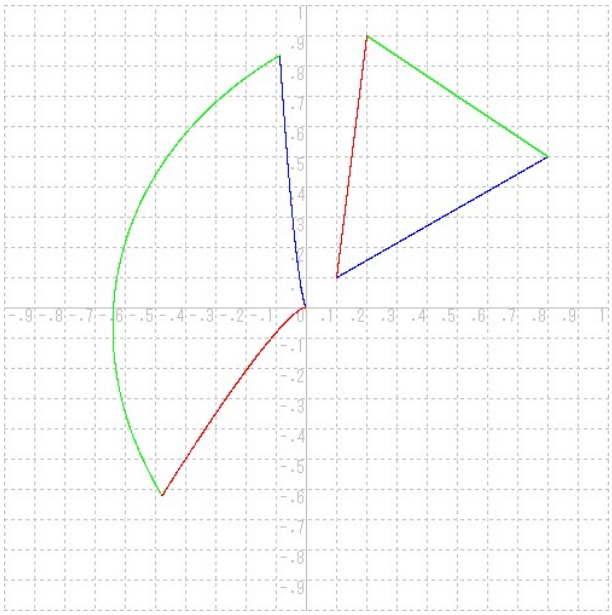
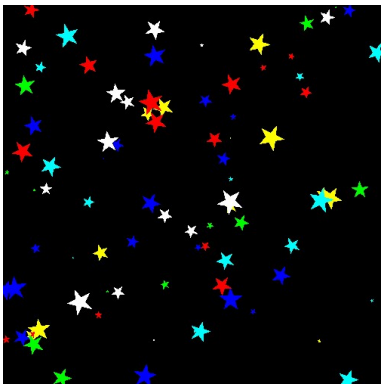


```
REM KOCH.BAS
OPTION ANGLE DEGREES
SUB right(t)
   LET direction=direction-t
END SUB
SUB left(t)
   LET direction=direction+t
END SUB
SUB walk(s)
   PLOT LINES: xp,yp;
   LET  xp=xp+s*COS(direction)
   LET  yp=yp+s*SIN(direction)
   PLOT LINES: xp,yp
END SUB
REM 端点の距離が n である Koch 曲線を描く
SUB koch(n)
   IF n>1 THEN
      CALL koch(n/3)
      CALL left(60)
      CALL koch(n/3)
      CALL right(120)
      CALL koch(n/3)
      CALL left(60)
      CALL koch(n/3)
   ELSE
      CALL walk(n)
   END IF
END SUB
LET  L=3^8
SET WINDOW -0.1*L,1.1*L,-0.1*L,1.1*L
REM 亀の方向と現在位置
LET  direction=0
LET  xp=0
LET  yp=0
CALL koch(L)
END
```

2　正則関数による写像

(準備中 ... すみません、時間切れです。)

```
REM 関数 w=z^3 で三角形がどういう図形に写像されるか
OPTION ARITHMETIC complex
LET  I=SQR(-1)
DECLARE EXTERNAL SUB segment
SET WINDOW -1,1,-1,1
SET LINE width 4
DRAW grid(0.1,0.1)
PRINT "複素平面上の三角形を関数 w=z^3 で写す"
PRINT "3 点の座標を入力してください。"
INPUT x1,y1
LET  z1=x1+y1*i
INPUT x2,y2
LET  z2=x2+y2*i
INPUT x3,y3
LET  z3=x3+y3*i
PRINT "z1=";z1
PRINT "z2=";z2
PRINT "z3=";z3
CALL segment(z1,z2,2)
CALL segment(z2,z3,3)
CALL segment(z3,z1,4)
END
REM 線分と線分の像
EXTERNAL SUB segment(z1,z2,c)
OPTION ARITHMETIC complex
DEF f(z)=z^3
SET LINE COLOR c
PLOT LINES
PLOT LINES : re(z1),im(z1);re(z2),im(z2)
FOR t=0 TO 1 STEP 0.001
   LET  z=(1-t)*z1+t*z2
   LET  w=z^3
   PLOT LINES: re(w),im(w);
NEXT t
END SUB
```

3 点の座標を入力して実行

複素平面上の三角形を関数 w=z^3 で写す
3 点の座標を入力してください。
? 0.1,0.1
? 0.8,0.5
? 0.2,0.9
z1=(.1 .1)
z2=(.8 .5)
z3=(.2 .9)