

J ウィンドウズ・グラフィックスへのいざない - Jできれいなグラフィックスを描こう -

西川 利男

Jはグラフィックスについても極めて強力である。言語としてのJのプログラミングは強力だがコンパクト過ぎ、ときにはトリッキーですらあり、BASICユーザには仲々なじめないようだ。しかし、Jのグラフィックスはすなおで伝統的であり、そんなことはない。通常 Windows のグラフィックスは VisualBasic で開発されるが、これが必ずしも易しくはない。フォームの作成、コードの記述など仲々面倒で、従来 DOS-BASIC になじんだユーザをとまどいさせている。これに比べるとJの Windows グラフィックス・プログラミングは Form Editor の使用で便利に行え、かつJのコードも見えて直接的で分かり易い。

今回はピックオーバの著「無限へチャレンジ」に紹介されていたラヌクロイドと呼ぶきれいなグラフィックス・パターンの作成を課題としてJの Windows グラフィックスの入門的解説を行う。

1. Jウィンドウズ・グラフィックスにおける3つの座標表示の方法

・ピクセル座標.....絶対座標

ディスプレイ画面のピクセル値による絶対座標

左上(0, 0)から右下(224, 200)への長方形内に表示

・Jグラフィックス座標...相対座標

Jの isigraph グラフィックパッケージにおける位置指定の座標

左下(0, 0)から右上(1000, 1000)への長方形内に表示

・数学座標.....相対座標

数学計算の値に合わせるため、例えば中心を(0, 0)に、右向きにx、上向きにyを適当なスケールでとる。

2. Jウィンドウズ・プログラミングのための Form Editor と wd コマンド

一般にウィンドウズのプログラムではユーザ・インターフェースとして直接キーインするのではなく、ボタン、エディットボックスなどアイコンにより入力する方式をとる。そのためプログラミングの最初の段階として、これらのアイコンを選んだり、割り付けたりすることが必要でこれはフォームと呼ばれる。Jではこれは Form Editor を用いることでごく簡単に行われる。

Jのスクリプト(=プログラム)作成ウィンドウ画面上で、[Edit]ツールバーから[Form Editor]を選んでクリックすると、Form Editor がただちに起動する。ウィンドウズ・オブジェクトの新規作成、ボタン、エディットボックスなどアイコンの選択、大きさ、位置

など是对話的に簡単に行われる。その編集も容易である。更にそのオブジェクトの上でダブルクリックすると、それに関するコード (= 処理プログラム) の作成、入力モードに入る。もちろんグラフィック・ウィンドウの作成も行われる。

J のウィンドウズ環境の特徴として、Form Editor で作成したフォームを J のコード定義として直接見ることが出来るので、細かい指定を意のままに修正することも可能である。

つまり J のウィンドウズの命令はコーディングとしては、wd (11!:)0 コマンドに文字列で操作内容を記述することで行われる。

ここでもっとも基本的な操作の wd コマンドの例を示す。

- ・親(parent)オブジェクトの操作

```
wd 'pc sample1;pn "My Graphics";'
```

親オブジェクトを sample1 として定義作成(create)し、"My Graphics" と表示する

```
wd 'pshow;'
```

親オブジェクトを画面に出力表示する

- ・子(child)オブジェクトの操作

```
wd 'xywh 100 50 34 12;cc ok button;cn "Run";'
```

ボタンの作成(create) - ピクセル位置(100, 50)から右へ 34 ピクセル、下へ 12 ピクセルの大きさのボタンを作成し、"Run" と表示する

```
wd 'xywh 20 20 120 100;cc mygraph isigraph;'
```

グラフィックス作成(create) - ピクセル位置(20, 20)から右へ 120 ピクセル、下へ 100 ピクセルの大きさの isigraph グラフィックス・ウィンドウを作成する

- ・J の isigraph による描画

```
wd 'glines 500 500 1000 500;'
```

J グラフィックス座標で、点(500, 500)から点(1000, 500)へ直線を描く

```
wd 'gpolygon 1000 500 500 1000 0 500 500 0;'
```

J グラフィックス座標で、点(1000, 500)、点(500, 1000)、点(0, 500)、点(0, 500)を結んで多角形(この場合は直線)を描く

```
wd 'gshow;'
```

グラフィックスを表示する

ここで注意すべきことはウィンドウ・オブジェクトの位置、大きさなどはピクセル座標で示し、グラフィックスでは J グラフィックス座標で表わすことの違いに注意！ なお、実際には Form Editor で自動的に生成されるので、ユーザはあまり意識することはないかもしれないが、しくみを知っておくことは必要である。

3. ラヌクロイド(Ranuncloid)グラフィックス

ピックオーバの著「無限へチャレンジ」の「創造の出現」の章の中にラヌクロイドなる極座標形式の包絡線で描かれるきれいなグラフィックスが紹介されている。

2つの点を次のようにするとき

$$P_0(x_0, y_0): x_0 = r \cos a, \quad y_0 = r \sin a$$

$$P_1(x_1, y_1): x_1 = r \cos b, \quad y_1 = r \sin b \quad ; \quad a, b \text{ はパラメータ}$$

これによって出来る線分 $P_0 P_1$ を角度 θ の $0^\circ \sim 360^\circ$ に対して次々と結んでいくことによって一種の包絡線群が現れ、これを一般にラヌクロイドと呼ぶ。ここでパラメータ a, b の値をいろいろと変化させると、ちょうどクモが糸で巣を張るようなきれいなパターンが現れる。 $a = 2, b = 2$ ではいわゆる心臓形(Cardioid)となる。このようなグラフィックスの作成を J でやってみる。

まず、ラヌクロイドの計算値を求める J の関数 chord は簡単に作られる。いまの場合 $r = 1$ とすると、 x, y の値の範囲は ± 1 となるので、この数学の値から J グラフィックスの値に変換することが必要になる。

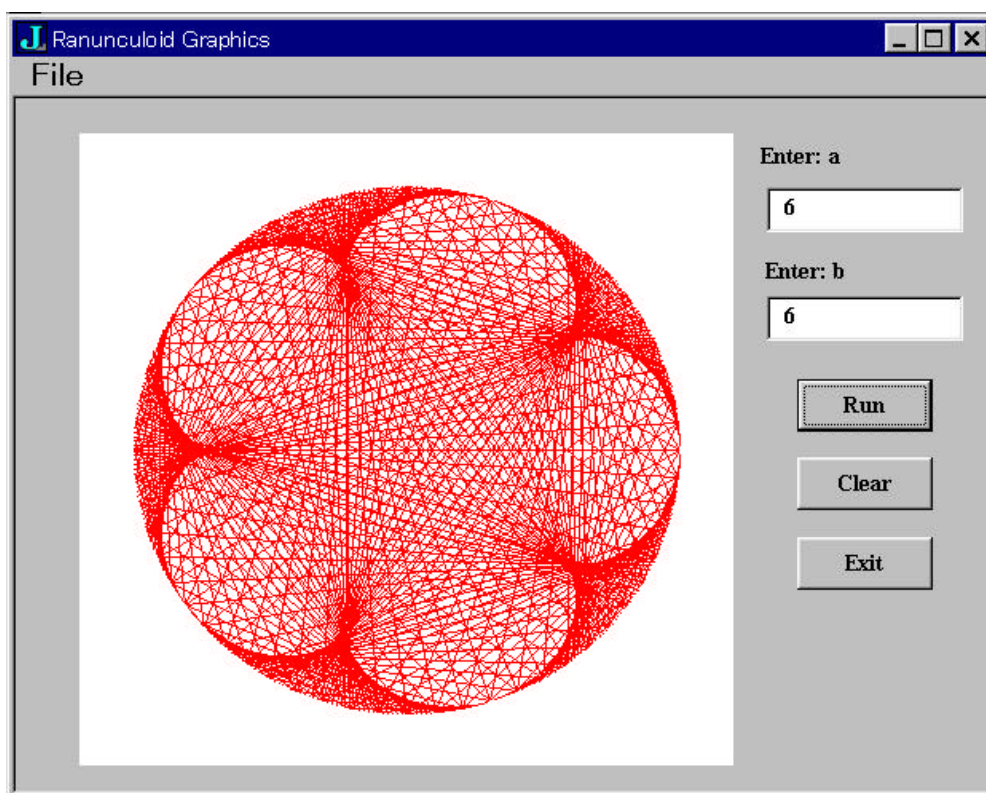
数学座標	J グラフィックス座標
-1.2	0
0	500
1.2	1000

これは J の次の関数で変換される。

```
adj =: %&1.2@(500&*(1.2&+))
```

このようにして得たグラフィックス座標の値を次々結んでいくことでラヌクロイド・グラフィックスが以下のように描かれる。パラメータ a, b の値をエディットボックスから入力し、ボタン Run をクリックすることで実行する。

プログラムの詳細は付録にあげた。パラメータ a, b を変えることで、いろいろきれいなパターンが現れる。実行例としてパラメータ $a = 6, b = 6$ に対するパターンを示した。



NB. Ranunculoid Graphs

NB. ピックオーバー著、一松信訳「無限へチャレンジしよう」森北出版(1997)

NB. 第9章 「創造の出現」 p.92-96

NB. base form

GRAPH=: 0 : 0

pc graph closeok;pn "Ranunculoid Graphics";

menupop "File";

menu new "&New" "" "" "";

menu open "&Open" "" "" "";

menusep ;

menu exit "&Exit" "" "" "";

menupopz;

xywh 196 64 34 12;cc ok button;cn "Run";

xywh 196 100 34 12;cc cancel button;cn "Exit";

xywh 16 8 164 144;cc ranunuc isigraph;

xywh 187 10 50 10;cc label static;cn "Enter: a";

xywh 188 20 50 11;cc a edit ws_border es_autohscroll;

xywh 188 36 50 10;cc label static;cn "Enter: b";

xywh 188 45 50 11;cc b edit ws_border es_autohscroll;

xywh 196 82 34 12;cc clear button;cn "Clear";

pas 6 6;pcenter;

rem form end;

)

run=: graph_run

graph_run=: 3 : 0

wd GRAPH

NB. initialize form here

wd 'set a 6'

wd 'set b 6'

wd 'gclear'

wd 'pshow;'

)

adj =: %&1.2@(500&*(1.2&+))

va =: 6

vb =: 6

```

graph_ok_button=: 3 : 0
wd 'grgb 255 0 0;'
wd 'gpen 1'
i =. 0
while. i < 360
  do. wd 'glines ', ": adj (va, vb) chord i
      i =. i + 1
  end.
wd 'gshow;'
wd 'pshow;'
)

```

```

graph_cancel_button=: 3 : 0
wd 'pclose;'
)

```

```

chord =: 3 : 0
:
('a';'b') =. x.
n =. y.
deg =. 1r180p1
t0 =. (2, 1)&o. deg * y.
t1 =. (2, 1)&o. (a, b) * deg * y.
t0, t1
)

```

```

graph_a_button=: 3 : 0
va =: ". a
)

```

```

graph_b_button=: 3 : 0
vb =: ". b
)

```

```

graph_clear_button=: 3 : 0
wd 'set a ', ' '
wd 'set b ', ' '
wd 'gclear'
)

```