

北斎の新形小紋帳から (その 1) J6 版

M.Shimura

JCD02773@nifty.ne.jp

2011 年 12 月 13 日

目次

1	小紋を描くー小紋のテクニック (1)	1
2	小紋のテクニック (2)-色の変化	12
3	いろいろな小紋のポリゴン	14
4	小紋のテクニック-図形の線形変換	20
5	ポリゴンと直線で描く	24
6	曲線の小紋	33

北斎の幾何デザイン

葛飾北斎は 60 歳過ぎに「新形小紋帳」という柳亭種彦の序文で始まる 60 ページ余の小刷を残しており (1824)、各ページの 2 個ずつの円内に様々な (江戸) 小紋が描かれている。種彦が「先生の筆頭より出でて古今に見も聞かざる紋をなす」としたためた北斎による 100 余りの幾何学模様の新形である。

画狂人北斎は 1760 年に両国本所の大江戸博物館の地で生まれ、1849 年永逝した。墓所は元浅草の誓教寺。富岳 36 景は 70 代から描き始めた。天保の変革 (1842) で種彦は亡くなり北斎も暫し江戸を離れた。

1 小紋を描くー小紋のテクニック (1)

北斎の井筒万字と六つ手万字でポリゴンを用いた小紋の基本テクニックを整理する。

1.1 キャンバス

- キャンバスは C.Reiter の dwin を用いる
`require '~addons/graphics/fvj3/dwin2.ijs'`
- 始点は左下で、終点は右上。

- サイズは指定でき、マイナスの値も用いることができるがデータにあわせて自動でセットはしてくれないので、指定サイズとなる xy の最小値と最大値を求めるツールを作成した

```
find_maxmin 4 5 calc_each_poly (<IM),<IMPARAM
_3 0 21 21 NB. min(x,y) , max(x,y)
```

```
_3 0 21 21 dwin 'IZUTU-MANJI'
```

- これを組み込んで自動でポップアップするようにもした。

```
_3 0 21 21 dwin 'IZUTU-MANJI'
```

- dwin は縦横比を自動調整しているので縦横比が大きい小紋のピースを連ねた画像ではピースの歪みが生じる。縦横比を補正してピースの形を保つような手法を *popup_dwin* に組み込んだ。辺縁部はカットするのでサイズ指定は大きめにすると良い

```
popup_dwin 4 5 calc_each_poly (<IM),<IMPARAM
```

```
find_center=: 3 : 0
```

```
NB. adjust canvas to square
```

```
tmp=: _2<\ 2 0 3 1 {; |. find_maxmin0 y NB. left bottom --> right top
```

```
Center=: -:@; |@:-/@|. L:0 tmp NB. center of gravity
```

```
Wide=: >. -:@:+/ ; 1r4&*@:-/ L:0 tmp NB. band
```

```
tmp1=:_2<\ Wide (+ , -@-) L:0 Center NB. range
```

```
ind=: ;*@>./ L:0 tmp1 NB. check _ or +
```

```
ANS=. <''
```

```
for_ctr. i.2 do.
```

```
if. _1 = ctr{ind do. PK=. >./ ; ctr{tmp1
```

```
else. PK=. <./ ; ctr{tmp1 end.
```

```
ANS=. ANS,<PK
```

```
end.
```

```
|. ; 2# L:0 }.ANS
```

```
)
```

1.2 カラーパレット

RGB 各 256 階調で 1600 万色になる。

	R	G	B	R	G	B	
Red	255	0	0	255	0	255	Magenta
Green	0	255	0	255	255	0	Yellow
Blue	0	0	255	0	255	255	Cyan
Black	0	0	0	255	255	255	White

後年の北斎はプロシャンブルーなどの輸入した合成絵の具を用いて明るい藍色を水や空に用いている。

光を重ねた色と絵の具を重ねた色には反対の性質がある。カラーブックには JIS 慣用色 269 が紹介されてい

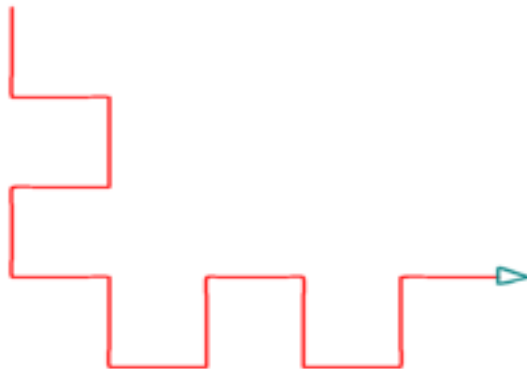
る。美術や印刷の世界の伝統色は趣があるようだ。日本の伝統色のカラーテーブルを紹介した本もいろいろあるので参照されたい。

Jのパッケージにも16色の色指定の他に *colortab.ijs* (150色) と *xwin.ijs* (500色) が含まれており、読み込めばカラーネームで使うこともできる。

Color	R	G	B	Color	R	G	B	Color	R	G	B
赤	190	0	50	緑	0	182	110	青	0	106	182
rose	219	53	97	green	0	154	87	blue	0	111	171
韓紅	230	75	107	常盤色	0	123	80	藍色	43	75	101
紅色	190	0	63	緑青色	77	129	105	瑠璃色	0	81	154
珊瑚色	255	127	143	emerald	0	164	116	marineblue	0	82	107
臙脂色	173	49	64	若竹色	0	163	126	navyblue	52	61	85
茜色	158	34	54	萌黄色	0	83	62	群青色	56	77	152
winered	128	39	63	若葉色	169	192	135	紺藍	53	53	115
scarlet	222	56	56	若草色	170	179	0	濃藍	34	53	70
chinesered	253	90	42					orientalblue	48	66	133
山吹色	248	169	0					江戸紫	97	72	118
橙/orange	239	129	15								
blond	222	178	95								
蜜柑色	235	132	0								

1.3 井筒万字の最初のピースの座標

- 一個のピースを方眼紙に描く
- 始点を決める(任意の点)。これが原点になる。
- ピースの左と下に x,y 軸を記入する
- 始点から左又は右回りに順にピースの頂点の座標を取っていく(一筆書き)
- ポリゴンの終点は始点の一個手前でよい。(自動で連結される)
- 井筒万字の最初の部分のデータをタートルグラフィックスで描いてみる。
始点は左上 $(,4)$ とする



```
require 'turtle'  
R180=: rt 180 fd 1  
R90=: rt 90 fd 1  
L90=: lt 90 fd 1  
  
show (R180,L90,R90,R90,L90,L90,R90,L90,L90,R90,R90,L90,L90)  
save '/temp/logo_izutu'
```

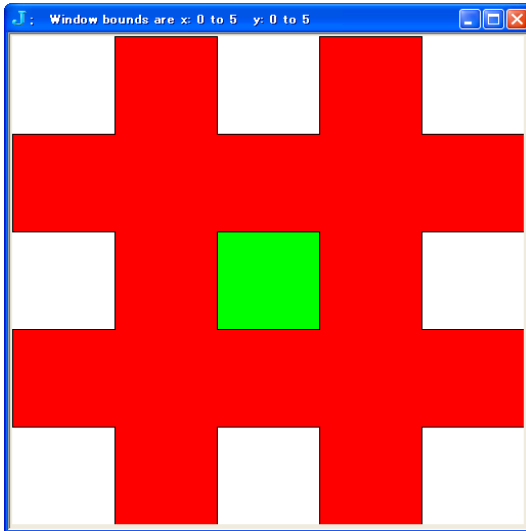
- J に入力するときはラミネート(,:)で縦型に。 $dpoly$ では見やすいようにポリゴンのデータは $(x\ y)$ を縦形とする。 $x\ y$ をペアにして2個ずつカンマで連結し、最後の一個をラミネートで連結すると縦形になる。

```

                                IM01,IM02
                                0 4
                                0 3
                                1 3
                                1 2
                                0 2
                                0 1
                                1 1
                                1 0
                                2 0
                                2 1
                                3 1
                                3 0
                                4 0
                                4 1
                                5 1
NB. 井筒組の万字
NB. first piece
IM01=:0 4,0 3,1 3,1 2,0 2,0 1,1 1,:1 0
IM02=:2 0,2 1,3 1,3 0,4 0,4 1,:5 1
IM03=:5 2,4 2,4 3,5 3,5 4,4 4,:4 5
IM04=:3 5,3 4,2 4,2 5,1 5,:1 4
IM=: IM01,IM02,IM03,IM04
NB. center square
IM2=: 2 3,2 2,3 2,: 3 3

```

- 最初のピース。2種類のポリゴンによる寄せ木細工またはジグゾウパズル



```

255 0 0 draw_dpoly IM
0 255 0 draw_dpoly IM2
NB. 井筒組の万字

```

1.4 小紋の同時描画-井筒万字を描く

同時描画 各ピースの座標をボックスで与え $dpoly$ を $(L:0)$ で用いるとループを用いなくて簡潔な手法で同時に描画できる。

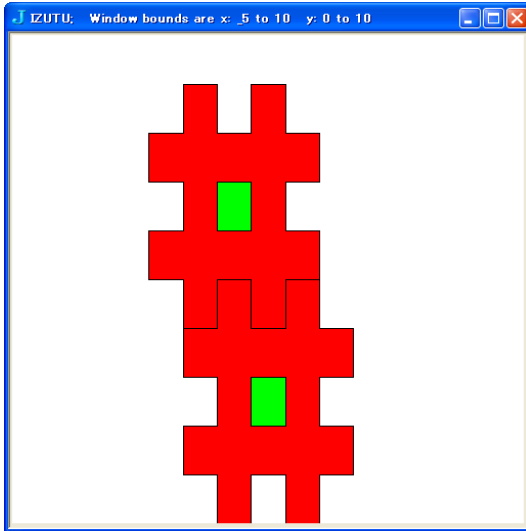
J はまだマルチ CPU での処理をサポートしていないので並列処理ではないが面倒なループを使用しない簡潔な同時処理ができる。

井筒万字のポリゴン 図形の始点 (左上 (0 4)) から外延のポイントを左回りに (x,y) を筆順になぞる。中の小さな正方形は別のポリゴンで描く

ボックスを用いて一度に描画する (同時処理)

組み木 上に一個積む (基準ポイント (0 4) からの移動量 (-1 4) を加える)。

(-1,4) は上の井筒の始点 (0 4) が左へ 1、上へ 4 上がったことを示す。これは方眼紙に図形を書いて値を得る。



```
_5 0 20 20 dwin 'IZUTU'
```

```
255 0 0 dpoly L:0 IM + "1 L:0 (0 0;_1 4)
```

```
0 255 0 dpoly L:0 IM2 + "1 L:0 (0 0;_1 4)
```

寄せ木 (又はタイル) の差分表 ボックスを用いて一度に計算するためのポイント表と差分表を作成する。方眼紙に幾つか寄せ木を描いて確認する。マスターピースは左下の (0,4) である。

小紋のフレーム

_3	16	1	17	5	18	9	19
_2	12	2	13	6	14	10	15
_1	8	3	9	7	10	11	11
0	4	4	5	8	6	12	7

差分表

IM_16

_3	12	1	13	5	14	9	15
_2	8	2	9	6	10	10	11
_1	4	3	5	7	6	11	7
0	0	4	1	8	2	12	3

左下がマスターピースで上と右に貼り付けていく。

原点は (0,4)

差分表はフレーム表から原点 (0,4) を差し引く。

小紋の差分パラメーターの定義 フレーム表や差分表をいちいち作るのは煩瑣なので差分パラメーターから生成する。

最初のピースの始点 ; X 軸方向の xy の差分; Y 軸方向の xy の差分

IMPARAM=: 0 4;4 1;_1 4

IM2PARAM=: 2 3;4 1;0 4

y 軸	_1 4		
	↑		
	0 0 ⇒	4 1	
	始点	X 軸	

y 軸	_1 4		
	↑		
	2 3 ⇒	4 1	
	始点	X 軸	

差分表の自動生成 差分パラメーターと小紋の数から差分表を生成する

- フレーム表の自動生成

```

4 4 mk_diff_sub0 IMPARAM
+-----+-----+-----+-----+
|_3 16|1 17|5 18|9 19 |
+-----+-----+-----+-----+
|_2 12|2 13|6 14|10 15|
+-----+-----+-----+-----+
|_1 8 |3 9 |7 10|11 11|
+-----+-----+-----+-----+
|0 4 |4 5 |8 6 |12 7 |
+-----+-----+-----+-----+

```

```

mk_diff_sub0=: 4 : 0
NB. Usage: 4 5 mk_diff_sub0 MCPARAM //10 5;_2 5;6
NB. x is size-of-matrix
NB. y is (<number of matrix),< base;diff x; diff y
'size_raw size_column'=. x
'base dfx dfy' =. y
X0=.|. { base +"1 (|. i.size_column) */ dfx
|: >{ L:0 X0 +"1 L:0 (|. i. size_raw) */ dfy
)

```

- 差分表の自動生成

```

4 5 mk_diff_sub1 IMPARAM
+-----+-----+-----+-----+
|_3 12|1 13|5 14|9 15 |
+-----+-----+-----+-----+
|_2 8 |2 9 |6 10|10 11|
+-----+-----+-----+-----+
|_1 4 |3 5 |7 6 |11 7 |
+-----+-----+-----+-----+
|0 0 |4 1 |8 2 |12 3 |
+-----+-----+-----+-----+

```

```

mk_diff_sub1=: 4 : 0
NB. Usage: 4 5 mk_diff_sub1 MCPARAM // 10 5;_2 5;6
NB. x is size-of-matrix (ex. 4 5) raw&column
NB. y is base;diff x; diff y
'base dfx dfy' =. y
tmp=. x mk_diff_sub0 y
tmp - L:0 base
)

```

小紋の各ピースので座標の計算 マスターピースと差分表から小紋の各ピースの座標を一度に計算する

```

4 5 calc_each_poly (<IM),<IMPARAM

calc_each_poly=: 4 : 0
NB. Usage: 4 5 calc_each_poly (<MC),<MCPARAM
NB. x is size-of-matrix
NB. y is (<piece),< parameter
'Piece Parameter'=. y
Piece + ("1) L:0 x mk_diff_sub1 Parameter
)

```

dwin での最小値と最大値 dwin に与える最小値と最大値の目安を取得し dwin に渡す

```

find_maxmin 4 5 calc_each_poly (<IM),<IMPARAM
_3 0 21 21

(0 0 255;4 5) draw_dpoly (<IM),<IMPARAM
(0 255 0 ;4 5) draw_dpoly_over (<IM2),<IM2PARAM

```

自動で小紋を描く 重ね描き用も作成する

- *draw_dpoly*

```
draw_dpoly=: 4 : 0
NB. x is color;size_of_matrix /raw & column(ex. 4 5)
NB. y is (<piece_data) , < diff_paramemter
NB. (255 0 255;4 5) draw_dpoly (<MC),<MCPARAM
'Color Size'=. x
tmp=. Size calc_each_poly y
popup_dwin tmp
Color dpoly L:0 tmp
)
```

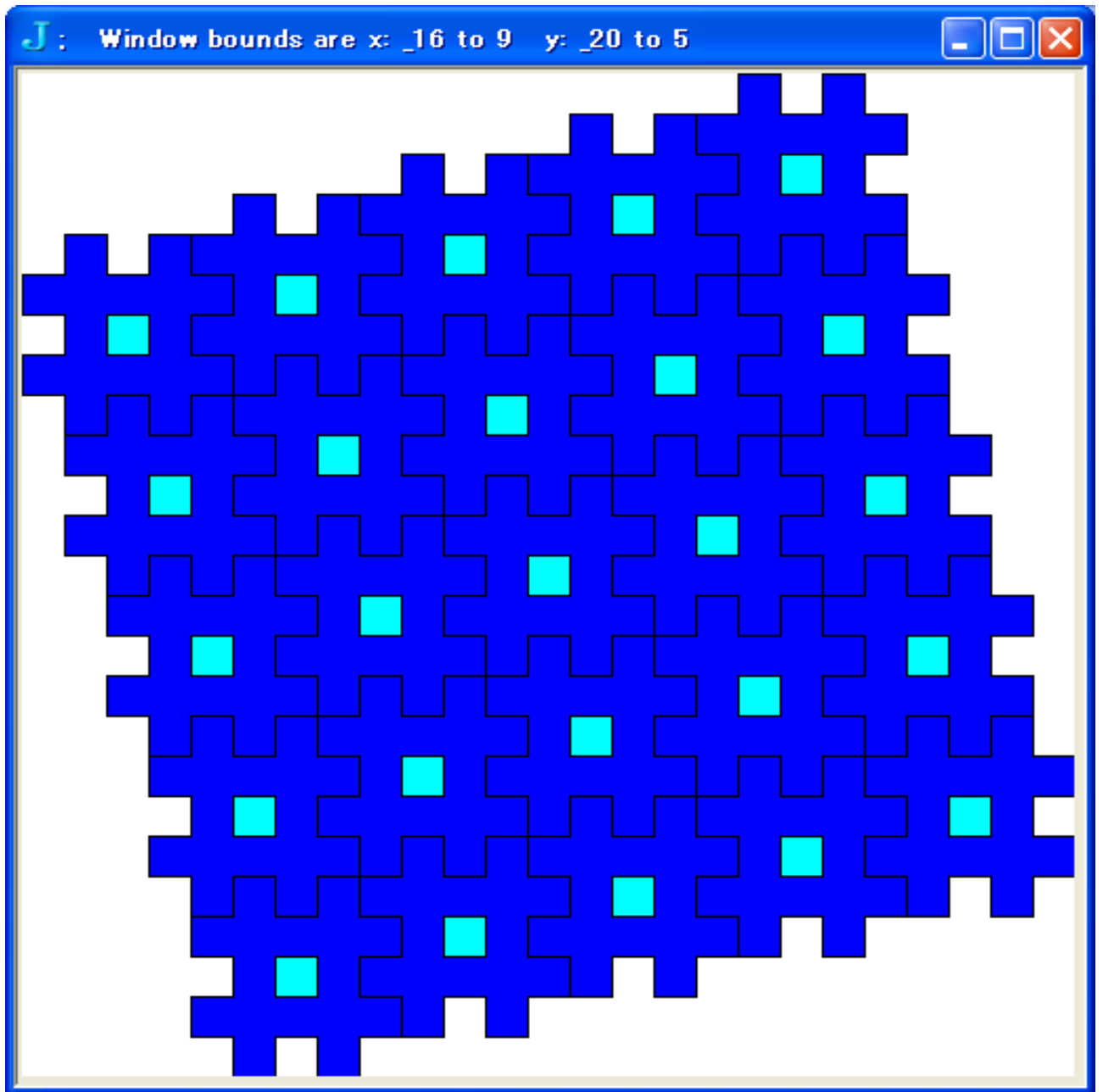
- *draw_dpoly_over*

```
draw_dpoly_over=: 4 : 0
NB. for overdrawing on shape/
NB. x y is same
'Color Size'=. x
tmp=. Size calc_each_poly y
Color dpoly L:0 tmp
)
```

一度に描く 色とサイズを指定して一度に描く手順のスク립トを作成した

```
hokusai_im=: 4 : 0
NB. (246 191 0;244 213 0 ;92 84 36) hokusai_im 6 7
NB. Color 0/1 graduation Color 2 is Center square
'Color0 Color1 Color2'=. x
Size=. y
((Color0;Color1);<Size) draw_dpoly_grad (<IM),<IMPARAM
(Color2;Size) draw_dpoly_over (<IM2),<IM2PARAM
)
```

```
(246 191 0;244 213 0;92 84 36) hokusai_im 10 10
```



1.5 六つ手万字をポリゴンで描く

六つ手万字は $A, B2$ 個のポリゴンを組み合わせる。回転と移動でもできるが 2 個作成したほうが手早い
最初のピースの座標 .

NB. 六つ手万字

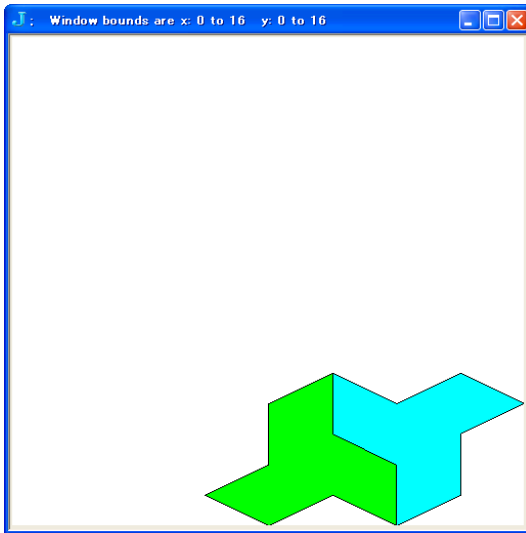
MM0=:10 5, 8 4, 8 2, 6 1, 8 0, 10 1, 12 0, 12 2, :10 3

MM1=: 10 5,10 3,12 2,12 0,14 1,14 3,16 4,14 5,:12 4

原点 (始点) は共用している

```
MM0 ,. MM1
10 5 | 10 5
 8 4 | 10 3
 8 2 | 12 2
 6 1 | 12 0
 8 0 | 14 1
10 1 | 14 3
12 0 | 16 4
12 2 | 14 5
10 3 | 12 4
```

最初のピースを描く .



```
0 0 16 16 dwin ''
```

```
0 255 0 dpoly MM0
```

```
0 255 255 dpoly MM1
```

ピースはこれを 90 度回転させたものと 2 組 (A,B) で構成されているが B は白抜きとすることもできる。

組み木用テーブル 図に方眼を重ね、フレームのため上と右の座標を求める。

4	20	10	19	16	18
6	15	12	14	18	13
8	10	14	9	20	8
10	5	16	4	22	3

8	10	
10	5	16 4

始点と差分 方眼紙に幾つかの絵を描くか、原画を眺めて原点の始点と各ピースの始点の差を読み取るここでは原点は A,B 共用である。

```
 2 5
  ↑
10 5 ⇒ 6 1
```

これは A,B 一緒である。

差分パラメーターを作成する .

```
MM0PARAM=: 10 5;6 _1;_2 5 NB. for automatic
MM1PARAM=: 10 5;6 _1;_2 5 NB. same MM0PARAM
```

小紋用の差分 小紋の各ピース用の差分を A,B について計算する

```
4 5 calc_each_poly (<MM0),<MM0PARAM
4 5 calc_each_poly (<MM1),<MM0PARAM
```

dpoly の最小値、最大値 dwin に与える最小値と最大値の目安を得る。(自動生成は最初の A で行う)

```
find_maxmin 4 5 calc_each_poly (<MM0),<MM0PARAM
0 _4 36 20
```

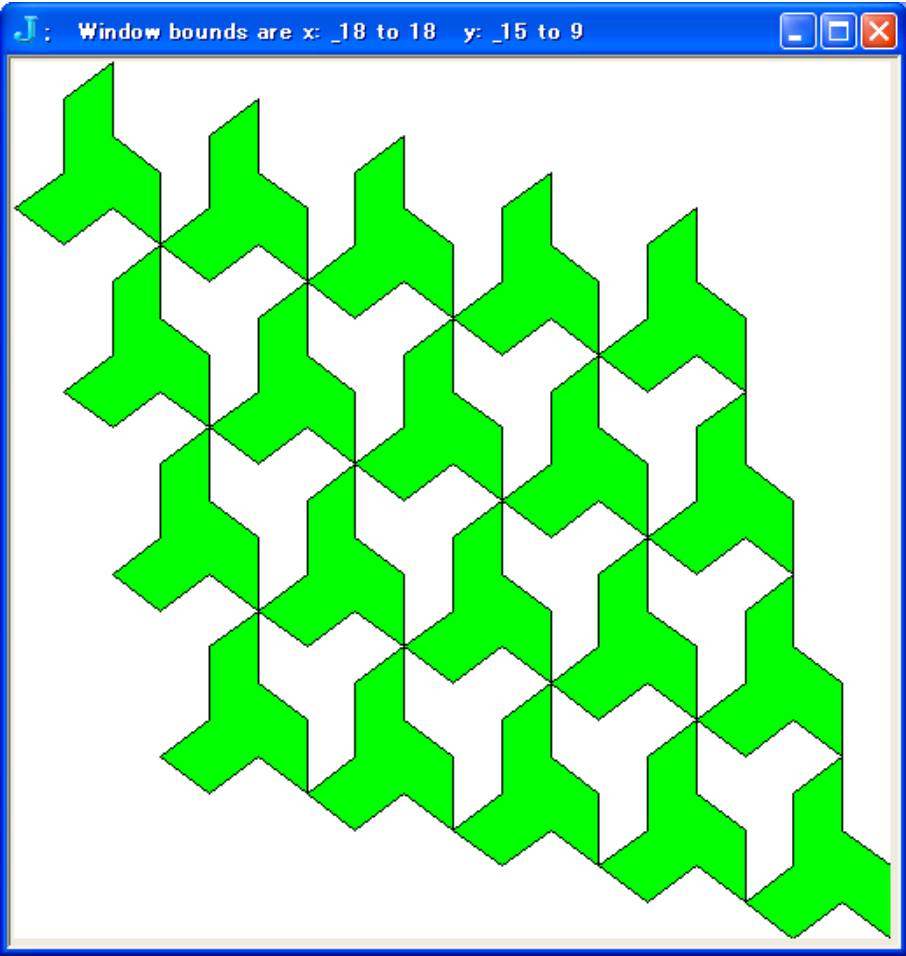
同時描画 .

白抜き

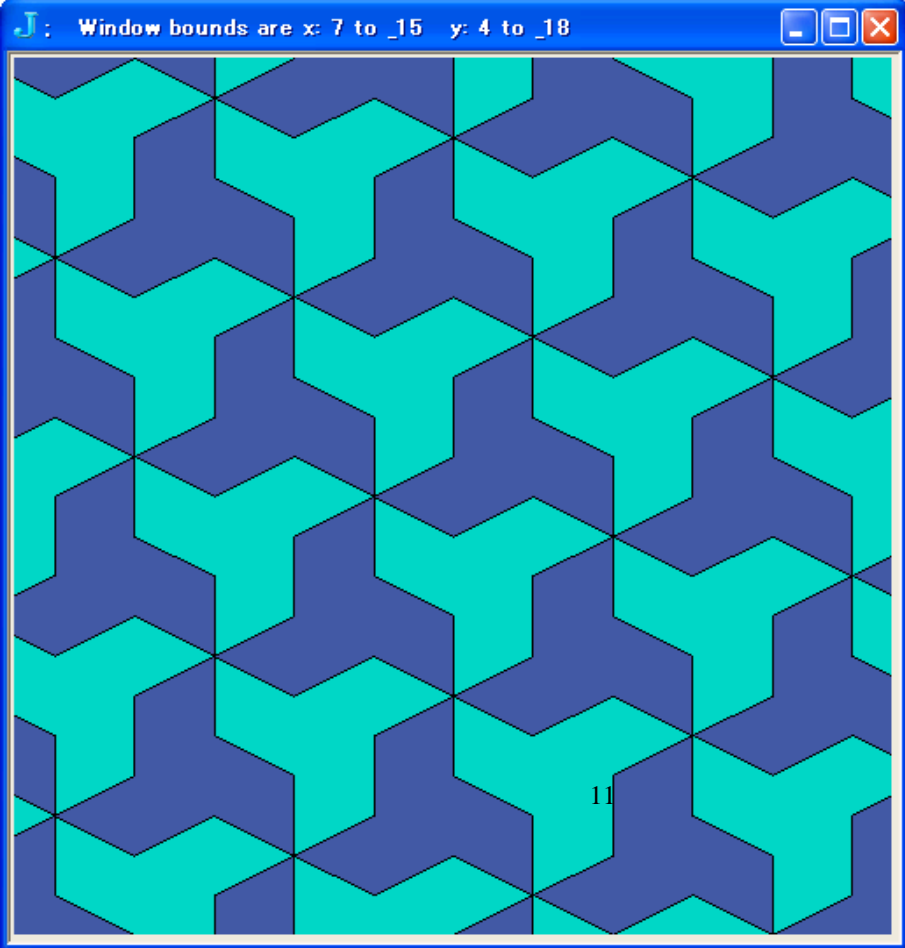
```
(0 0 255 ;4 5 )draw_dpoly (<MM0),<MM0PARAM NB. A 単独では白抜き
```

2色

```
(67 71 173; 67 90 160) hokusai_mm 6 7
```



ht



2 小紋のテクニーク (2)-色の変化

六つ手万字のように A,B2 つの図形がある場合は色による変化を付けやすいが、井筒万字のような主体部分が J 種類の図形は同じ色だとのっぺりとするので色で変化を付けたい。

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

ここでは小紋の個数は縦横の大きさを任意に取れるようにしている。よく眺めると赤と黒は右上、左下の斜めの線に沿って並んでいる。これは J の *Oblique* 機能によって斜め線に分類できる。

```
</. i. 5 4
+-----+-----+-----+-----+-----+
|0|1 4|2 5 8|3 6 9 12|7 10 13 16|11 14 17|15 18|19|
+-----+-----+-----+-----+-----+

</. i. 5 5
+-----+-----+-----+-----+-----+
|0|1 5|2 6 10|3 7 11 15|4 8 12 16 20|9 13 17 21|14 18 22|19 23|24|
+-----+-----+-----+-----+-----+
```

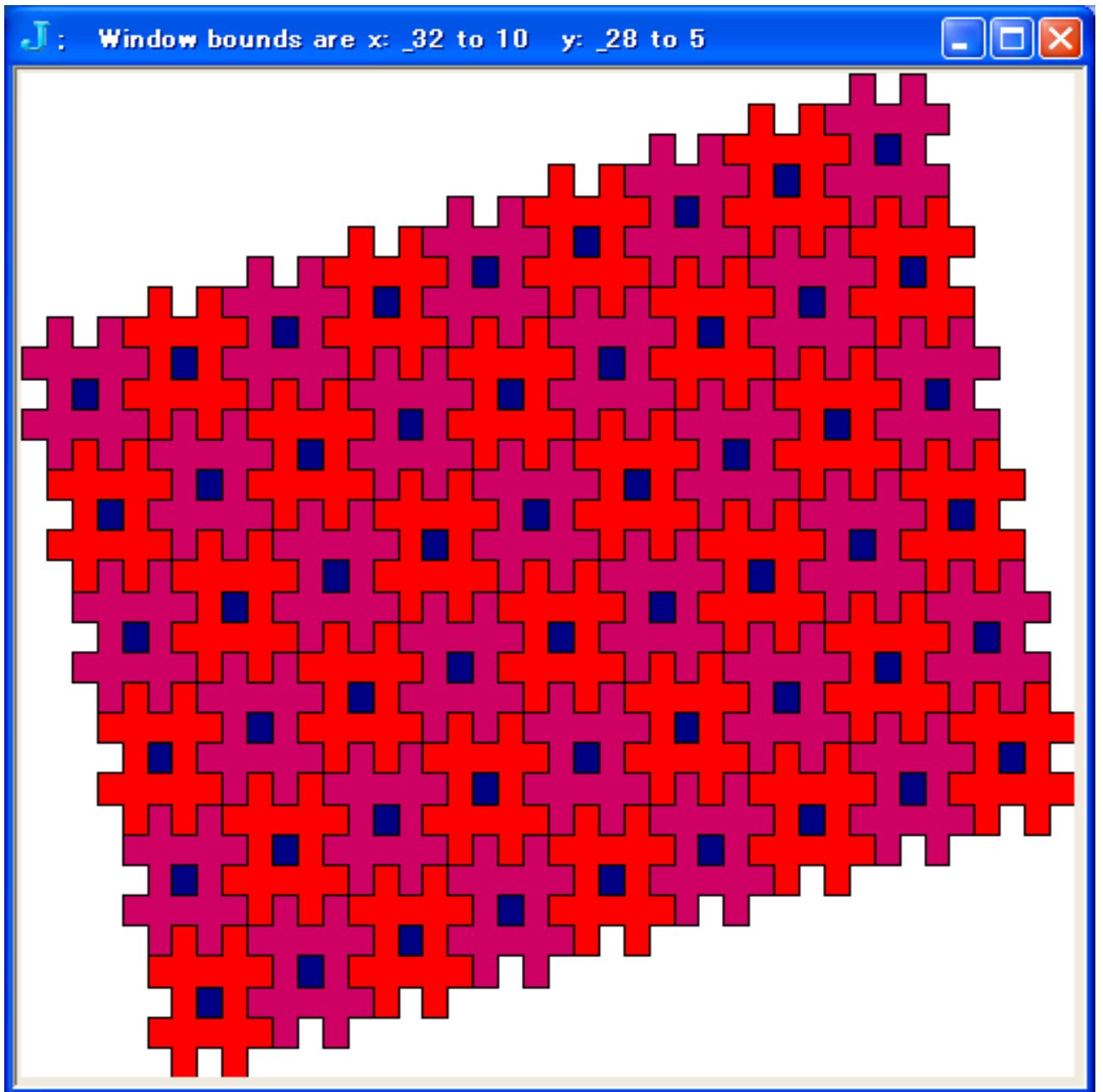
これを交互に紅白に分類し、この数値を指標として差分計算済みのデータを区分して描画関数に渡せばよい。次の簡潔なスクリプトで分離できる

```
index_separate=: 3 : 0
NB. for graduation 2 colors
NB. index_separate 3 5
ind=. i. # Oblic=. </. i. y NB. using oblique
tmp0=.;(-.2|ind)# Oblic
tmp1=.;(2|ind)#Oblic
tmp0;tmp1
)

index_separate 5 5
+-----+-----+-----+-----+-----+
|0 2 6 10 4 8 12 16 20 14 18 22 24|1 5 3 7 11 15 9 13 17 21 19 23|
+-----+-----+-----+-----+-----+
```

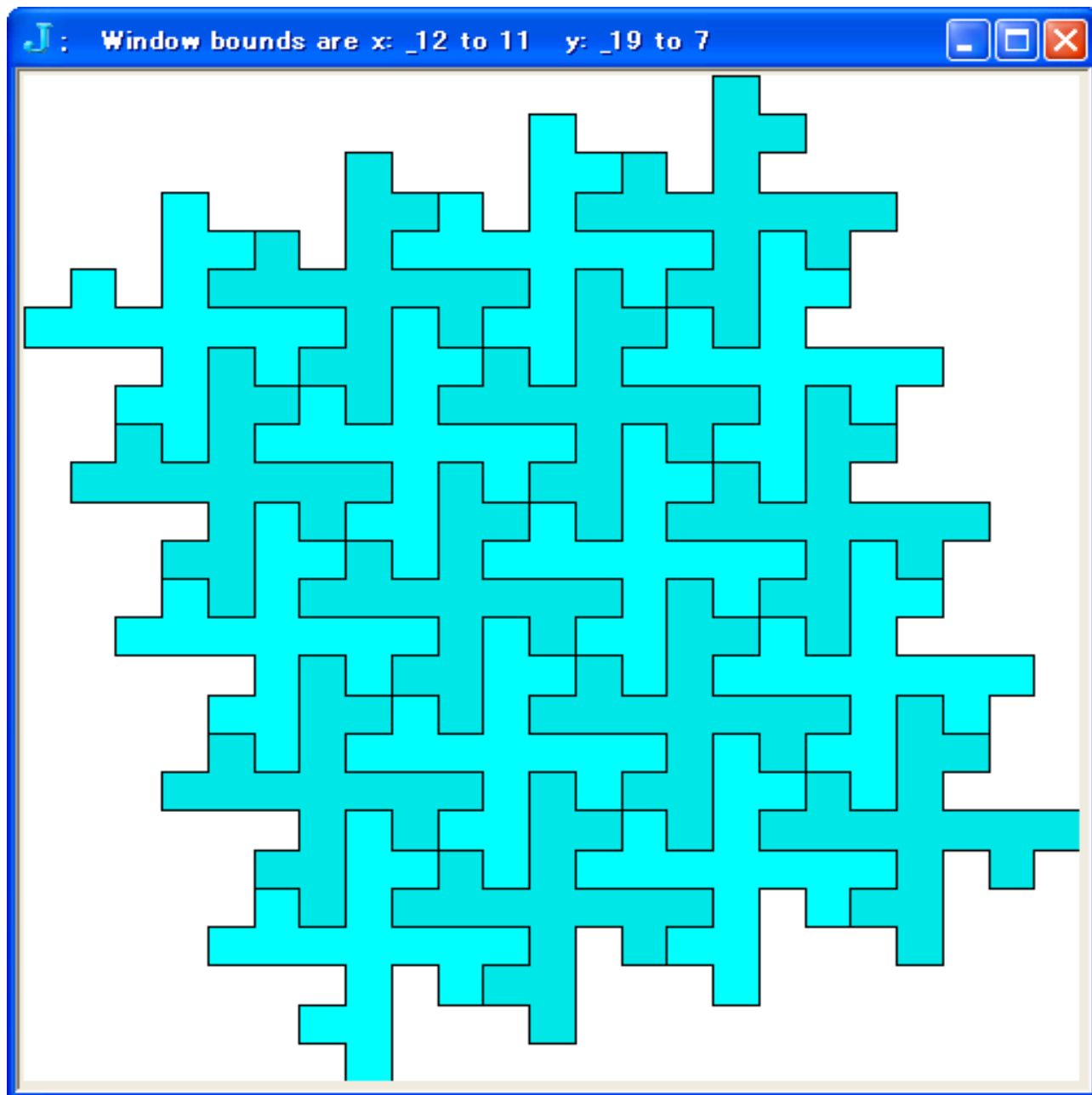
一連の作業を行うため *draw_dplot_grad* を作成した

```
(( 0 0 255 ; 0 0 240);< 4 5) draw_dpoly_grad (<IM),<IMPARAM  
( 0 200 0;4 5) draw_dpoly_ove (<IM2),<IM2PARAM
```



3 いろいろな小紋のポリゴン

3.1 賽形



最初のピースの入力 .

NB. xx saigata

S0=: 0 3,3 3,3 2,2 2,2 1,3 1,3 0,: 4 0

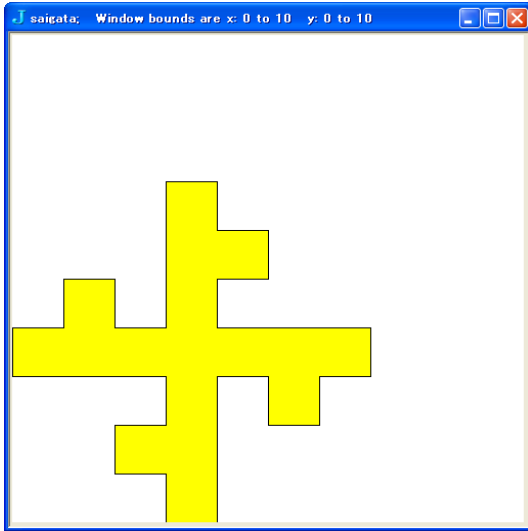
S1=: 4 3,5 3,5 2,6 2,6 3,7 3,:7 4

```

S2=: 4 4,4 5,5 5,5 6,4 6,4 7,:3 7
S3=: 3 4,2 4,2 5,1 5,1 4,:0 4
S=: S0,S1,S2,S3

```

最初のピース .



```
255 255 0 draw_dpoly0 S
```

ピースの差分テーブル .

小紋のフレーム

```

4 5 mk_diff_sub0 SPARAM
+-----+-----+-----+-----+-----+
|12 6|11 10|10 14|9 18 |8 22 |
+-----+-----+-----+-----+
|8 5 |7 9 | 6 13 |5 17 |4 21 |
+-----+-----+-----+-----+
|4 4 |3 8 | 2 12 |1 16 |0 20 |
+-----+-----+-----+-----+
|0 3 |_1 7 |_2 11|_3 15|_4 19|
+-----+-----+-----+-----+

```

```

_1 7
  ↑
  0 3 ⇒ 4 4

```

```
SPARAM=: 0 3;_1 4;4 1
```

draw_dpoly_grad .

```

hokusai_s=: 4 : 0
NB. Usage:(67 71 173; 67 90 160) hokusai_s 6 7
'Color0 Color1'=. x
Size=. y
((Color0;Color1);<Size) draw_dpoly_grad (<S),<SPARAM
)

```

3.2 万字つなぎ

A,B2 個の図形である。

最初のピース .

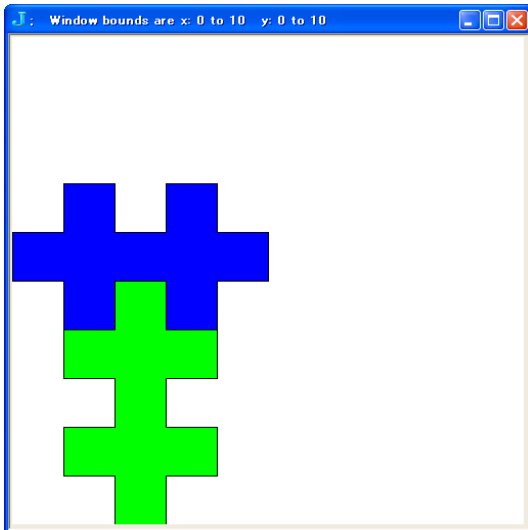
NB. Manji Tunagi

MT0=: 1 4,1 3,2 3,2 2,1 2,1 1,2 1,2 0,3 0,3 1,:4 1

MT0=:MT0,4 2,3 2,3 3,4 3,4 4,3 4,3 5,2 5,: 2 4

MT1=: 0 6,0 5,1 5,1 4,2 4,2 5,3 5,3 4,4 4,4 5,:5 5

MT1=: MT1,5 6,4 6,4 7,3 7,3 6,2 6,2 7,1 7,:1 6



0 0 10 10 dwin ''

0 255 0 dpoly MT0

0 0 255 dpoly MT1

差分パラメータ .

小紋フレーム

1 10
↑
1 4 ⇒ 4 1

差分

MTOPARAM=: 1 4;3 _3; 0 6

0 6
↑
1 4 ⇒ 3 _3

draw_dpoly .

小紋フレーム

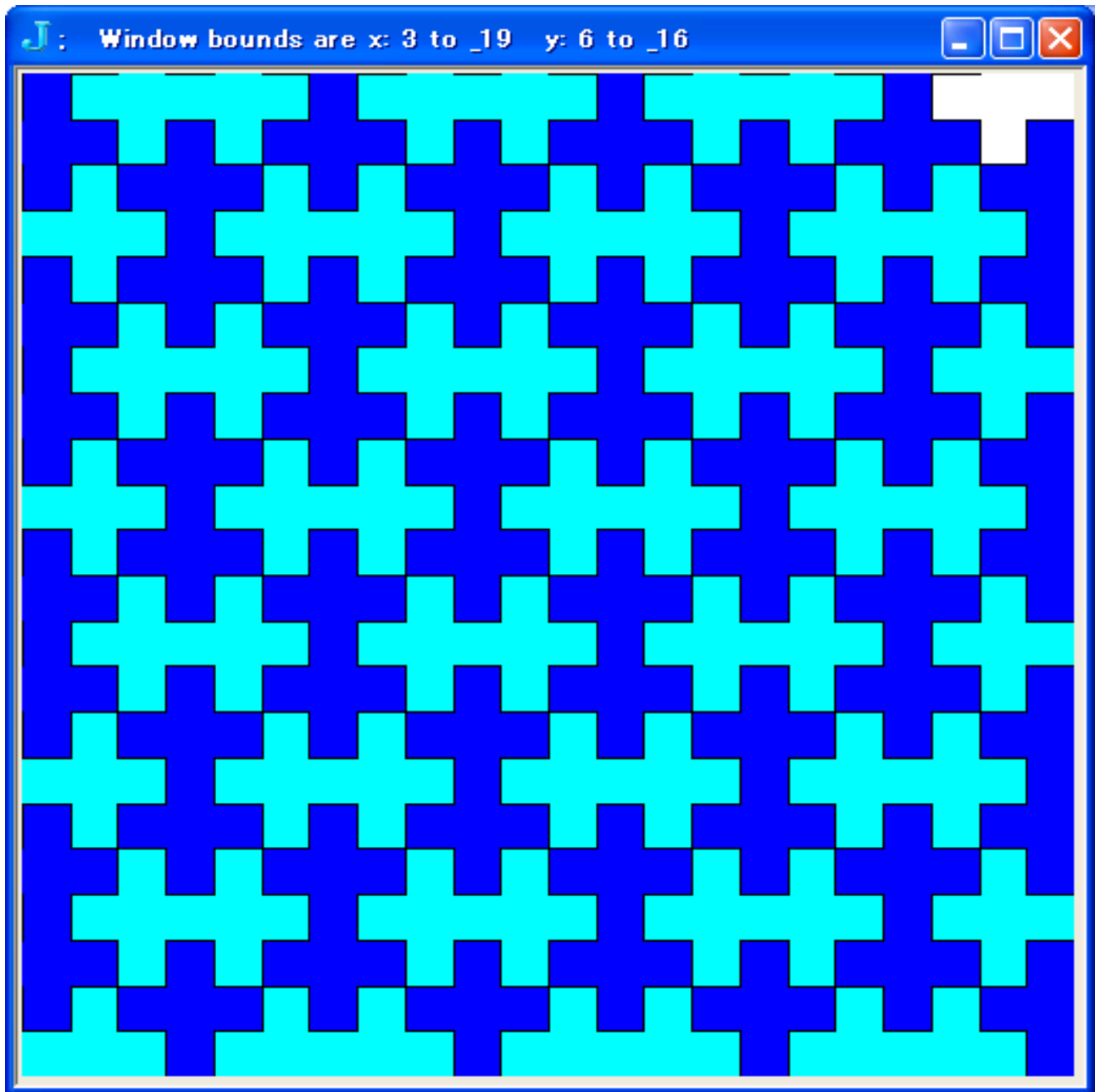
0 12
↑
0 6 ⇒ 3 3

差分

MTIPARAM=: 0 6;3 _3;0 6

0 6
↑
0 6 ⇒ 6 0

(144 93 84;230 75 107) hokusai_mt 6 7



3.3 万字菱

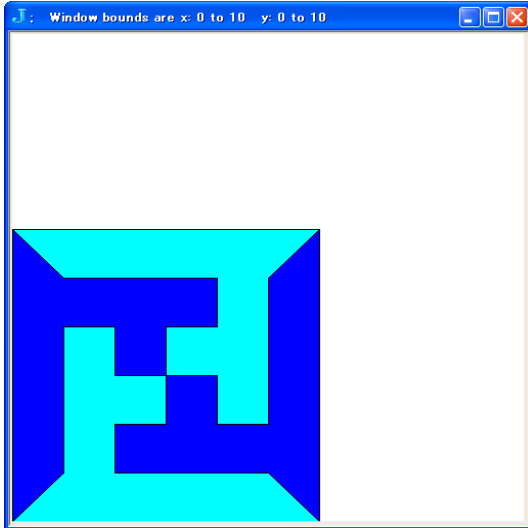
最初のピース .

4個の万字で菱形を作っている。1点でつながった2個のピースは一個のピースとして扱えるのでA,Bの2個として扱う。

MC=: 3 3,4 3,4 2,5 2,5 5,6 6,0 6,1 5,4 5,4 4,3 4,: 3 3

MC=: MC,3 3,3 2,2 2,2 1,5 1,6 0,0 0,1 1,1 4,2 4,2 3,:3 3

MCPARAM=: 3 3;6 0;0 6



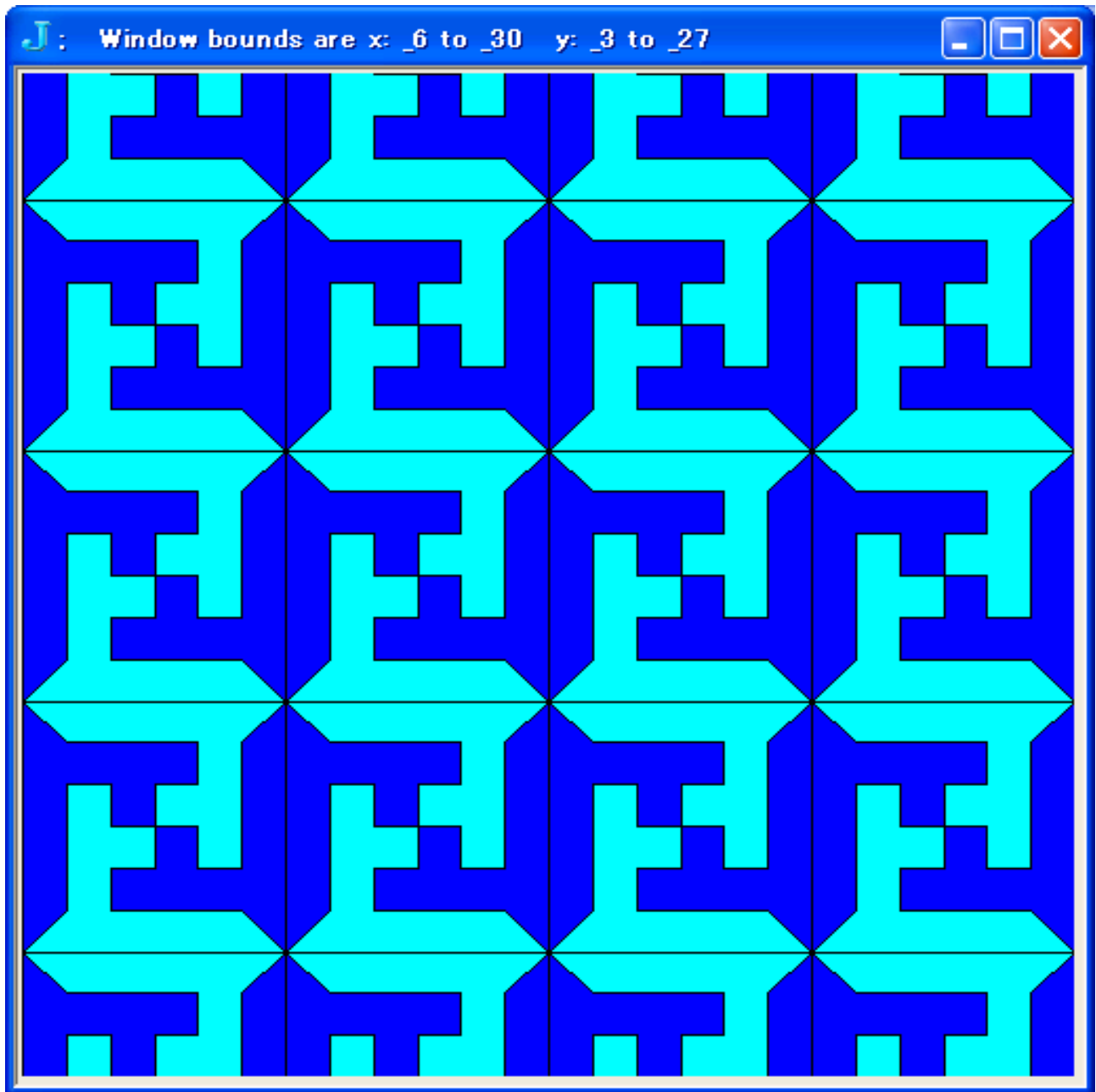
0 0 6 6 dwin 'manjihishi'
0 255 255 dpoly MC0

差分パラメータ .

MCPARAM=: 3 3;6 0;0 6

差分

0 6
↑
3 3 ⇒ 6 0



```

hokusai_mc=: 4 : 0
NB. (0 164 116 ;169 192 135) hokusai_mc 6 7
'Color0 Color1 '= . x
'Size'= . y
(Color0;Size) draw_dpoly (<MC0),<MCOPARAM
(Color1;Size) draw_dpoly_over (<MC1),<MCOPARAM
(0 0 0;Size) draw_dline_over (<MC0),<MCOPARAM
(0 0 0;Size) draw_dline_over (<MC1),<MCOPARAM

```

)

4 小紋のテクニク-図形の線形変換

4.1 Homogeneous Coordinates

図形の伸縮、回転、移動は次のように行える。3×3のマトリクスを用いる場合は *Homogeneous Coordinates* と呼ばれる。本来 3 次元の回転であるが z 軸を固定した場合に相当し 2 時平面で作用する。

- 葉書を串で突き刺して回した場合に相当。
x を 1 にして展開すれば X 軸が、y が 1 では y 軸が、z に 1 を持ってくる と z 軸が回転する。今回は 2 次なので z を 1 にする。

$$\begin{matrix} x & . & . \\ . & y & . \\ . & . & z \end{matrix}$$

- 図形データは (x,y) に z 軸分の 1 を加えて計算し、描画のときに削除する
- 計算は内積演算 mp=: +/ . *
- ポリゴンやラインの x,y のデータには z 軸にあたる列に 1 を加えて計算してグラフィックスに渡す前に落とす

Script 同時変換の Script

```
elongm=: 3 : '(y,1)* =i.3'
rotm=: (cos, sin,0:),(-@sin,cos,0:),: 0: ,0:,1:
NB. rotm by C.Reiter
transm=: 3 : '(=i.2), y,1'
mp=: +/ . * NB. inner products
```

回転 反時計回りに回転する。

$$(x, y, 1)_{new} = (x, y, 1) \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
rotm 1r4p1
0.707107 0.707107 0
_0.707107 0.707107 0
0 0 1
```

伸縮 横に 2 倍、縦に 3 倍に伸ばす。縮小はマイナスで

$$(x, y, 1)_{new} = (x, y, 1) \begin{pmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
elongm 2 3
2 0 0
0 3 0
0 0 1
```

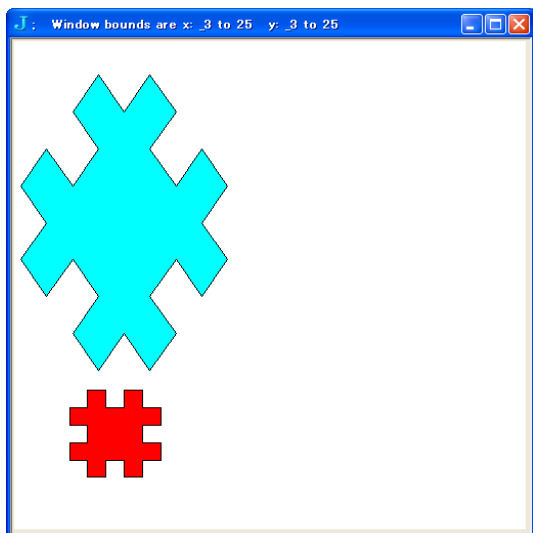
移動 x,y=(3 4) に移動

$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix} \quad \begin{matrix} \text{elongm 2 3} \\ 1 \ 0 \ 0 \\ 0 \ 1 \ 0 \\ 3 \ 4 \ 1 \end{matrix}$$

全部まとめて .

$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix}$$

```
a=. (rotm 1r4p1) mp (elongm 2 3) mp transm 3 4
1.41421 2.12132 0
_1.41421 2.12132 0
      3      4 1
```

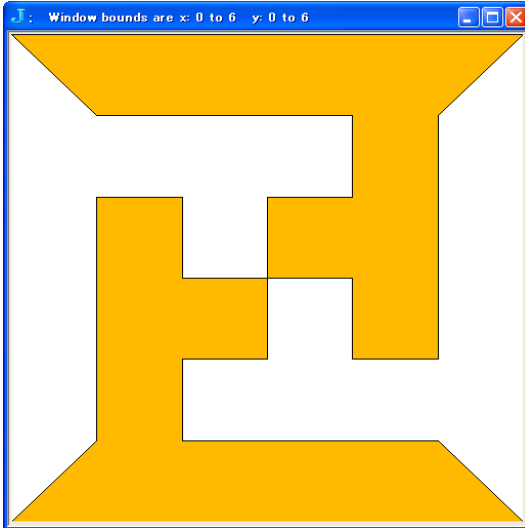


```
_3 _3 25 25 dwin ''
255 0 0 dpoly IM
0 255 255 dpoly a1=. }:"1 (IM,.1)mp a
```

4.2 万字菱を傾ける

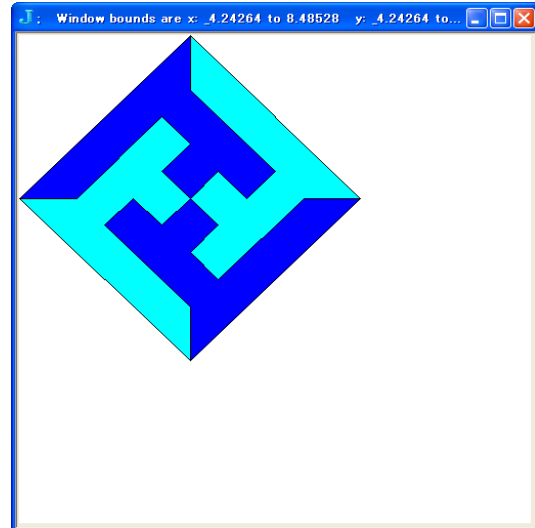
万字菱と回転 左回りに $\frac{1}{4}\pi$ 回転させる

255 187 0 draw_dpoly0 MC0
 NB. 向日葵色



4 5 度回転 左回り (キャンパスの都合で小さく見えるが縮小はしていない)

```
0 0 255 draw_dpoly0 }:"1 (MC0,.1) mp rotm 1r4p1
0 255 255 dpoly }:"1 (MC1,.1) mp rotm 1r4p1
```



小紋のフレームを傾ける 小紋は直交フレームで作ってある。これを指定ラジアン分傾けたフレームを作る

- 直交フレーム

```
a=. 4 5 mk_diff_sub0 MC0PARAM
+-----+-----+-----+-----+
|3 21|9 21|15 21|21 21|27 21|
+-----+-----+-----+-----+
|3 15|9 15|15 15|21 15|27 15|
+-----+-----+-----+-----+
|3 9 |9 9 |15 9 |21 9 |27 9 |
+-----+-----+-----+-----+
|3 3 |9 3 |15 3 |21 3 |27 3 |
+-----+-----+-----+-----+
```

- 直交フレームの各ベクトルをを回転行列で回転させ、左下の原点との差分を取る

```
(3 4;1r4p1) mk_diff_sub1_diagonal MC0PARAM
+-----+-----+-----+-----+
|8.48528 8.48528|4.24264 12.7279 |0 16.9706      |_-4.24264 21.2132|
+-----+-----+-----+-----+
|4.24264 4.24264|0 8.48528      |_-4.24264 12.7279|_-8.48528 16.9706|
+-----+-----+-----+-----+
|0 0          |_-4.24264 4.24264|_-8.48528 8.48528|_-12.7279 12.7279|
+-----+-----+-----+-----+
```

ピースを回転させる .

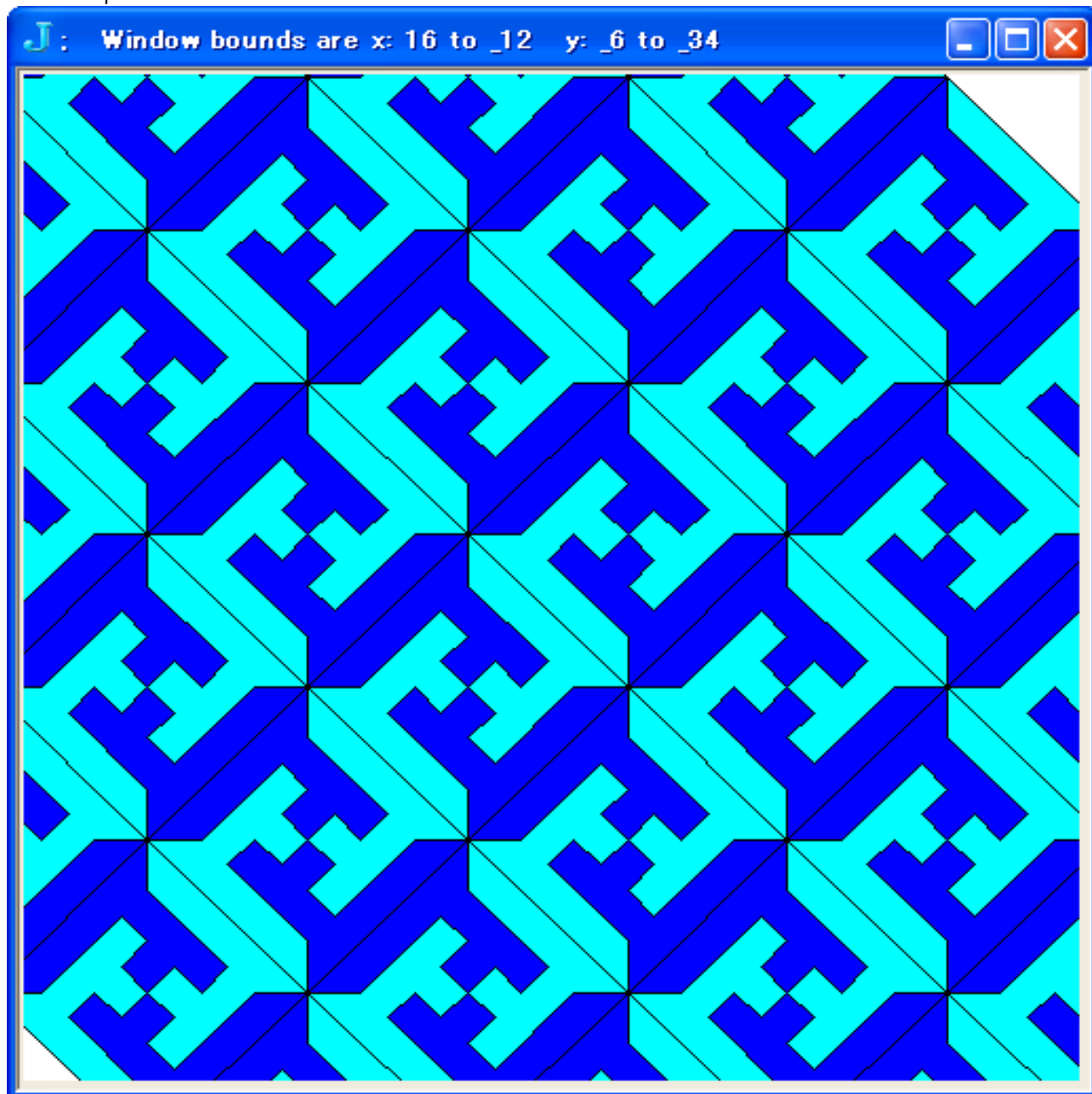
```
}:"1 ( MC0,.1) mp rotm 1r4p1
```

小紋の各座標を計算する 回転させたピースから差分座標を引く

```
(0 0 255;6 7;1r4p1) draw_dpoly_diagonal (<MC0),<MC0PARAM
```

```
(0 255 255;6 7;1r4p1) draw_dpoly_over_diagonal (<MC1),<MC0PARAM
```

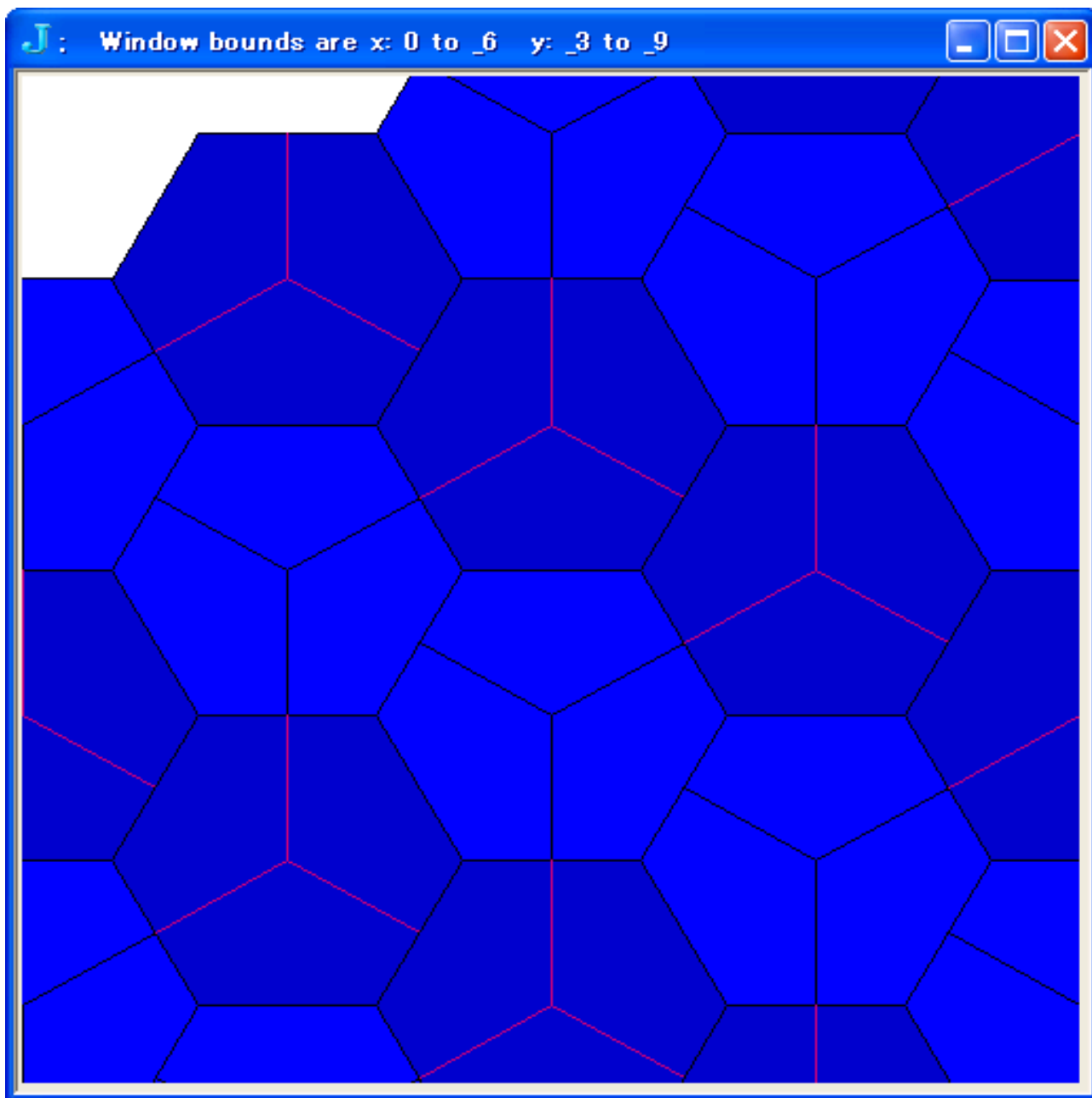
1r4p1 は $\frac{1}{4}\pi$ ラジアン



5 ポリゴンと直線で描く

5.1 ひょうれつ麻の葉

```
((0 0 255;0 0 200 );<5 5) draw_dpoly_grad (<HA0),<HA0PARAM  
((0 0 0 ; 200 0 120);<5 5) ha_line_over ''
```



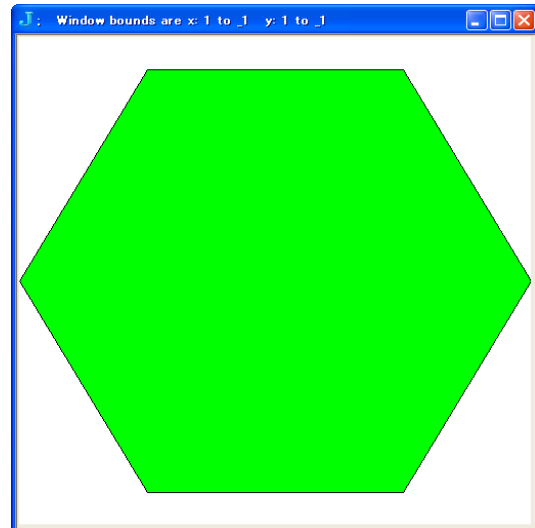
六角形 正多角形を描くには *J* のイディオムが便利である。正多角形の数字を変えれば何角形でも描ける

が、定規とコンパスと言う古典問題ではない。さらに、正多角形の各頂点の座標が得られるので便利である。

0 255 0 draw_dpoly0 HA0

```
] HA0=: +. r. 2p1*(i.6)%6 NB. hexagon
  1      0
0.5    0.866025
_0.5   0.866025
_1    1.22465e_16
_0.5   _0.866025
 0.5   _0.866025
```

始点は中央右

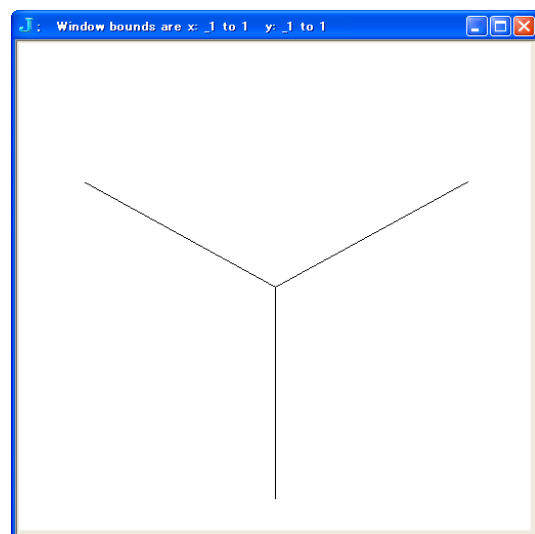
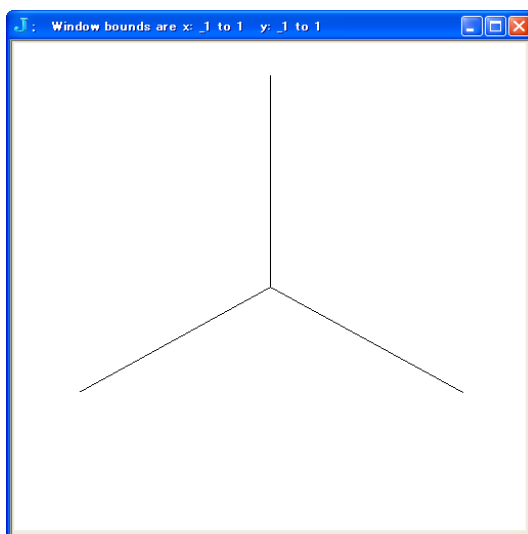


直線 直線は *dline* で描く。座標データの形式はポリゴンと同じであるが、ラインは終点から始点まで自動で連結してはくれない。

乱れ麻を直線で描く Y と逆 Y の 2 種類である

HAL0, HAL1 値は方眼の図から計算や読み取りで求めた

HAL0		HAL1	
0	0	0	0
0	0.866025	0	_0.866025
0	0	0	0
0.75	_0.43317	0.75	0.43317
0	0	0	0
_0.75	_0.43317	_0.75	0.43317



始点は Y の中心である

差分 次の表から始点との差を採る.

$$\begin{array}{cc|c} 1 & 1.7325 & \\ \uparrow & & \\ 1 & 0 & \Rightarrow 2.5 \quad 0.866 \end{array}$$

HA0PARAM=:1 0 ; 1.5 0.866;0 1.732

$$\begin{array}{cc|c} 1 & 1.7325 & \\ \uparrow & & \\ 0 & 0 & \Rightarrow 1.5 \quad 0.866 \end{array}$$

HAL1PARAM=:HAL0PARAM=: 0 0 ; 1.5 0.86625;0
1.7325

麻の葉を描く 北斎の Y、逆 Y の組み合わせから規則性は分からなかったのでグラデュエーションのパーツを流用して手順を書き出した。Y、逆 Y はグラデュエーションと一致するが、乱数を入れるともっと粋になるのではないかな

hokusai_ha=: 4 : 0

NB. (144 93 84;230 75 107;0 0 0;144 93 84) hokusai_ha 6 7

'Color0 Color1 Color2 Color3'=. x

Size=. y

tmp0=.Size calc_each_poly (<HAL0),<HAL0PARAM

tmp1=.Size calc_each_poly (<HAL1),<HAL0PARAM

'Ind0 Ind1'=. index_separate Size

'Gr0 Gr1'=. (<Ind0 { ,tmp0),<Ind1{,tmp1

((Color0;Color1);<Size) draw_dpoly_grad (<HA0),<HA0PARAM

Color2 dline L:0 Gr0

Color3 dline L:0 Gr1

)

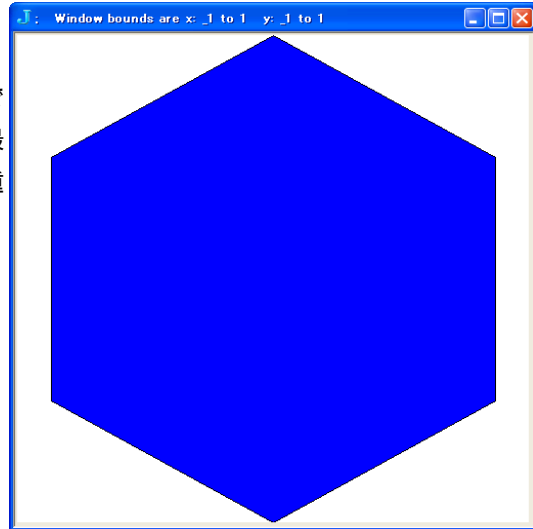
5.2 麻の葉つくし

正六角形 前の六角形を流用するが、頂角がある。

```
_1 _1 1 1 dwin ''
0 0 255 dpoly AT
```

AT=: (4|.|.|.|"1 HA0),|.1{HA0
dline でも 6 角形を書くために始点と終点を繋いで
 おこう。ポリゴンは図を重ねると重なった部分は最
 後の図(色)が表示される。絵の具のように塗り重
 ねの痕跡は残らない

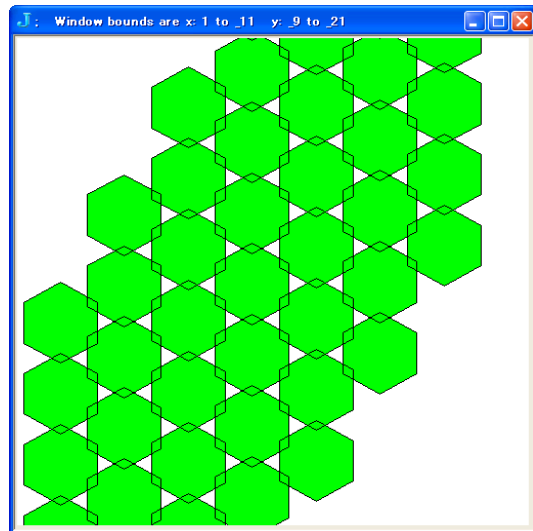
```
clean AT
0.866025 0.5
      0 1
_0.866025 0.5
_0.866025 _0.5
      0 _1
0.866025 _0.5
0.866025 0.5
```



dline 塗り重ねて消えた輪郭線を再度描く
 差分 図を多く描いて調整した

ATPARAM=: 0.866 0.5; 0 1.75; 1.5 2.65
 差分パラメーターの構成

Y	軸		
1.5	2.65		
	↑		
0.866	0.5	⇒	0 1.75
始	点		X 軸

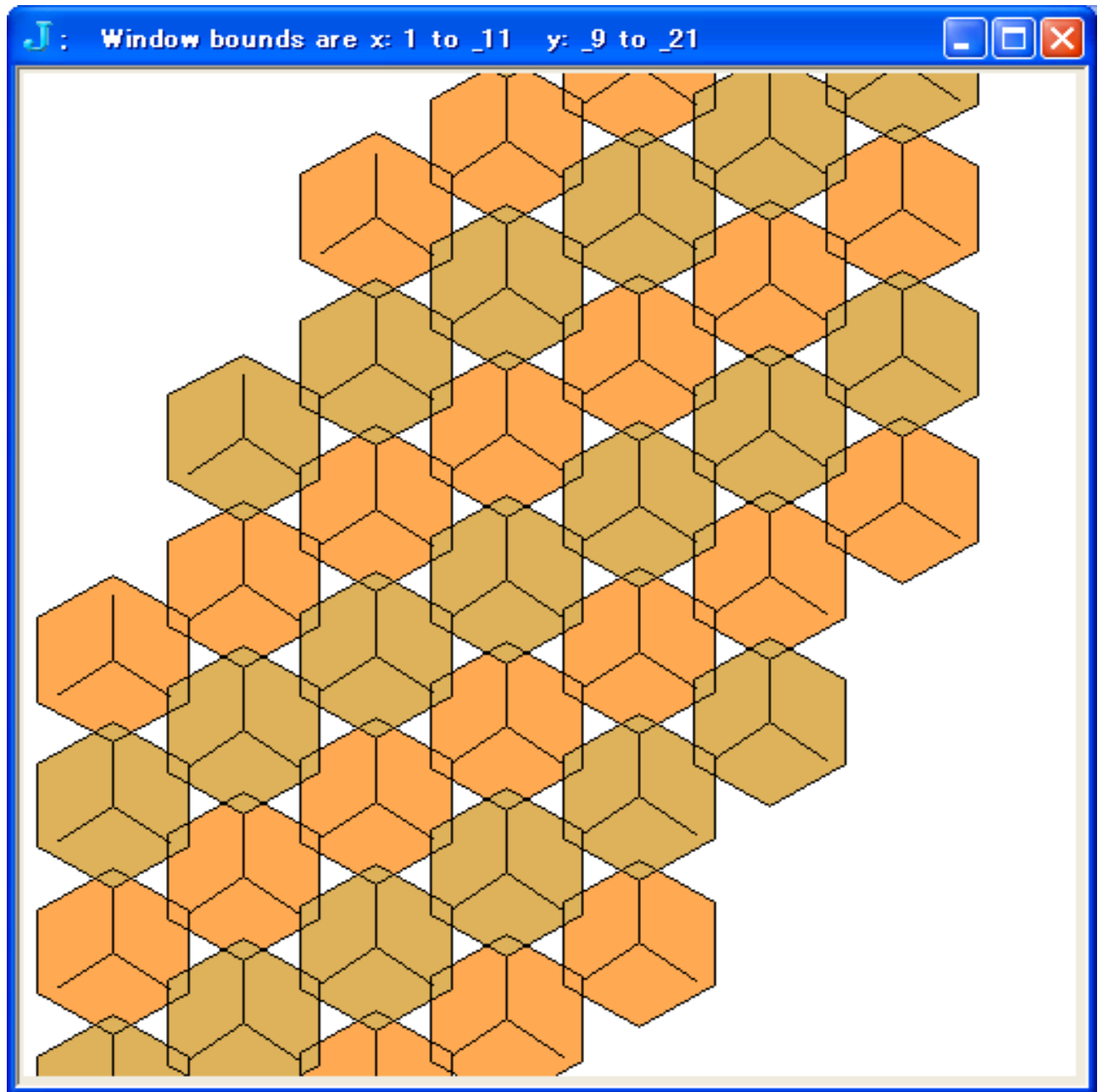


Yの線 draw_dline_over で重ね描きする

```
ATL=: 0 0 , _0.65 0.45, 0 0, 0 _0.75, 0 0 ,: 0.65 0.45
```

```
(144 93 84;230 75 107;0 0 0) hokusai_at 6 7
```

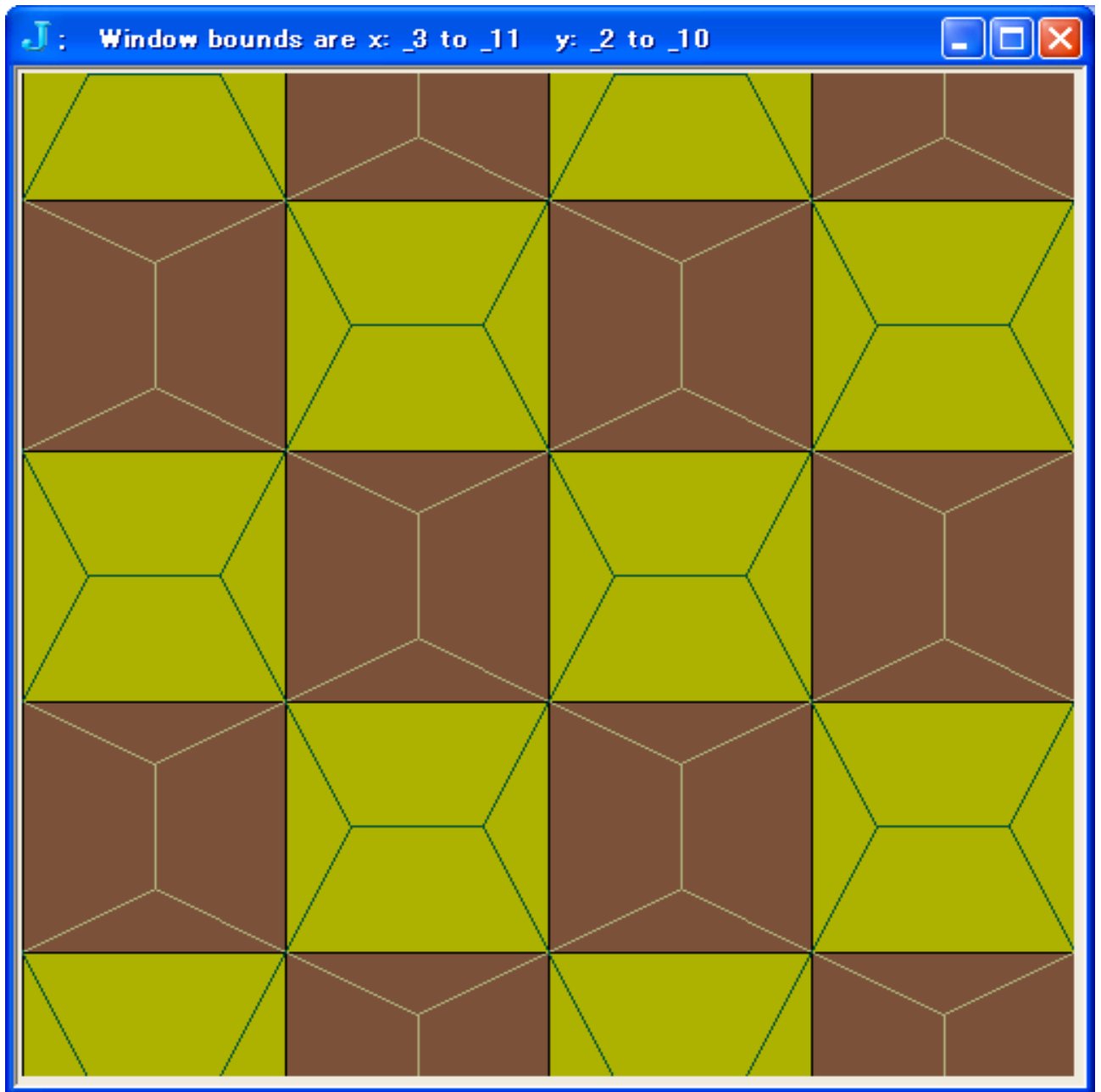
色は山吹色とブロンド



5.3 八つ手万字

```
((0 83 62;170 179 0;170 179 0;0 83 62);<7 8) ym_grad_over ''
```

線も色指定できるようにし、パラメーターが多いので手順を単独のスク립トに書いた。色は若竹と萌黄を使っている。バックグラウンドの色が濃い場合は *dline* の色を白 (255 255 255) などの色指定にしないと濃い色に埋没する



正方形 .

NB. -----

NB. 八手麻の葉

NB. 8 hands Hemp leaves

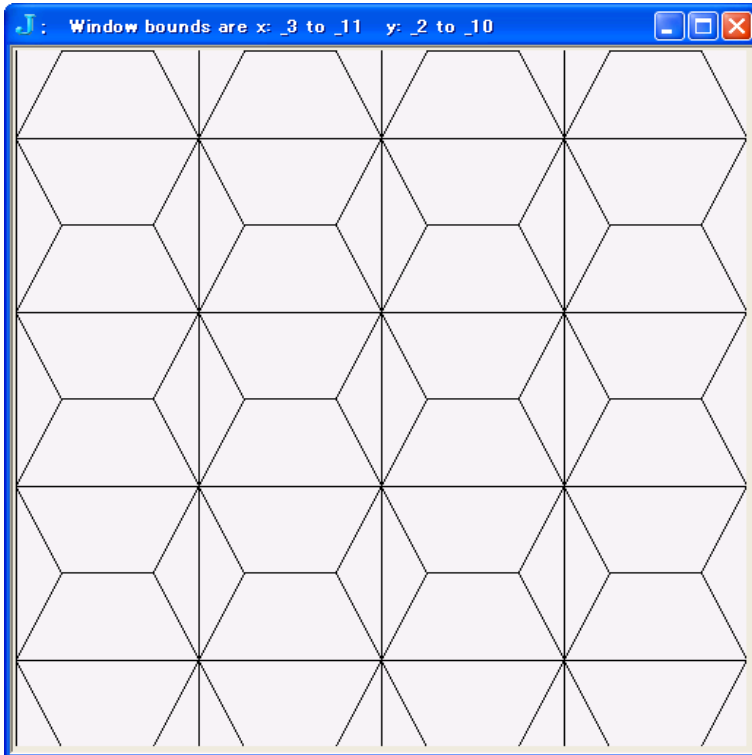
YM0=: 1 1, _1 1, _1 _1 ,1 _1,: 1 1

線 クロスの線と付加する線。YML_2 は YML_1 を 90 度回転させるための x,y を入れ替える。

```

YML0=: 0 0, _1 1, 0 0, _1 _1, 0 0, 1 _1 , 0 0, 1 1, : 0 0
YML1=: 0 0, 0 0.5, 1 1, 0 0.5, _1 1, 0 0.5 , 0 _0.5, 1 _1, 0 _0.5, : _1 _1
YML2=: |."1 YML1

```



差分 4の図形とも正方形の重心(0,0)の積み重ね分

```
YMPARAM=: 0 0; 2 0; 0 2
```

手順 手順は引数が多いので script に書き出した

```

hokusai_ym=: 4 : 0
NB. (0 83 62; 170 179 0; 170 179 0; 0 83 62) hokusai_ym 7 8
NB. color 0 1 -> dpoly 2 3 -> dline
NB. color 2 3 is reverse of 0 1 or 0 0 0 /255 255 255
'Size'=. y
'Color0 Color1 Color2 Color3'=. x
tmp0=.Size calc_each_poly (<YML0),<YMPARAM
tmp1=.Size calc_each_poly (<YML1),<YMPARAM
tmp2=.Size calc_each_poly (<YML2),<YMPARAM
'Ind0 Ind1'=. index_separate Size
'Gr0 Gr1'=. (<Ind0 { ,tmp0),<Ind1{,tmp0
'Gr2 Gr3'=. (<Ind0 { ,tmp1),<Ind1{,tmp2
popup_dwin tmp0
Color0 dpoly L:0 Gr0

```

```

Color1 dpoly L:0 Gr1
Color2 dline L:0 Gr2
Color3 dline L:0 Gr3
)

```

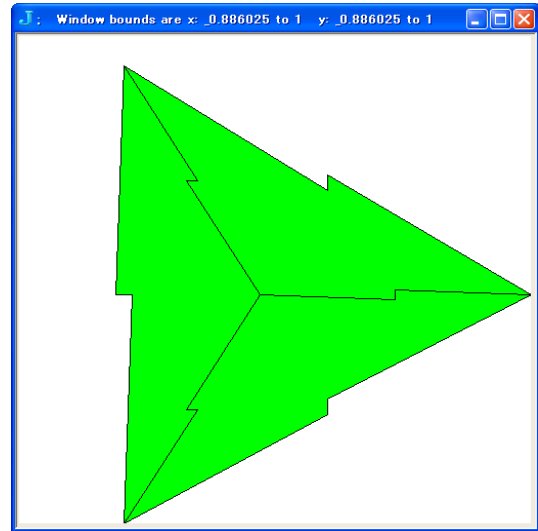
5.4 松川麻の葉

三角形 3 角形の 2 つのピース。各辺に稲妻を入れる

```

MA0, .MA1
_0.5 0.886025| 1 1.732
_0.47 0| 0.22 1.299
_0.53 0| 0.28 1.299
_0.5 _0.886025| _0.5 0.886025
0.25 _0.463| 0.22 0.433
0.25 _0.403| 0.28 0.433
1 0| 1 0
0.25 0.463| 0.97 0.866
0.25 0.403| 1.03 0.866
_0.5 0.886025| 1 1.732

```



Y 字稲妻の線分 三角形の重心を起点として一筆書きするため、行きつ戻りつしている。数値は図形からざっと求めたもので精密ではない。

color3 color3 は線の色。バックが濃い場合は白抜き (255 255 255) などが映える

差分パラメータ 2 の三角形と 2 の Y 字の差分。

```

MA0PARAM=: _0.5 0.866025;1.5 0.866025;0 1.73205
MA1PARAM=: 1 1.73205;1.5 0.866025;0 1.73205
MAL0PARAM=: 0 0;1.5 0.866025;0 1.73205
MAL1PARAM=: 1 1.73205;1.5 0.866025;0 1.73205

```

```
hokusai_ma=: 4 : 0
```

```
NB. (169 192 135;151 166 30;104 126 82) hokusai_ma 6 7
```

```
NB. y is Size
```

```
'Color0 Color1 Color2'=. x
```

```
Size=. y
```

```
tmp0=.Size calc_each_poly (<MA0),<MA0PARAM
```

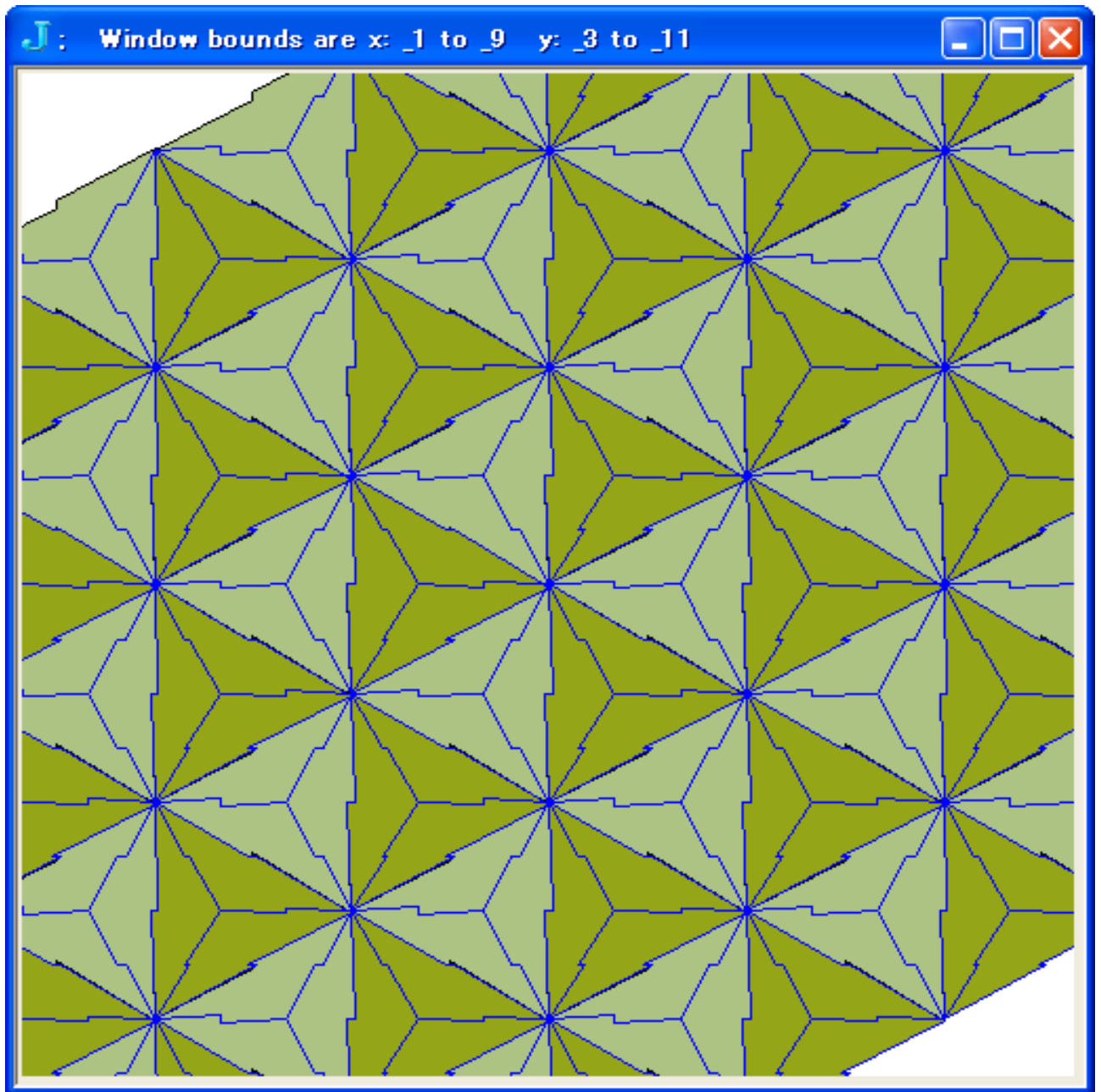
```
tmp1=.Size calc_each_poly (<MA1),<MA1PARAM
```

```
tmp2=.Size calc_each_poly (<MAL0),<MAL0PARAM
```

```
tmp3=.Size calc_each_poly (<MAL1),<MAL1PARAM
```

```
'Ind0 Ind1'=. index_separate Size
'Gr0 Gr1'=. (<Ind0 { ,tmp0),<Ind1{,tmp0
'Gr2 Gr3'=. (<Ind0 { ,tmp1),<Ind1{,tmp2
popup_dwin tmp0
Color0 dpoly L:0 Gr0
Color1 dpoly L:0 Gr1
Color1 dline L:0 Gr1
Color2 dline L:0 tmp2
Color2 dline L:0 tmp3
)
```

(169 192 135;151 166 30;0 0 255)hokusai_ma 6 7



6 曲線の小紋

(to be continued)

References

永田生慈監修解説「北斎の絵手本(3)」 岩崎美術社 1986
ファー・インク編「カラーハンドブック」MDN Corporation 2010

J602 J701 はトロントから *DL* 出来ます

<http://www.jsoftware.com>

スクリプトは次から *DL* できます

<http://japla.sakura/ne.jp> の *symposium 2011*