

江戸小紋でオリンピックとパラリンピックマークを描く

SHIMURA Masato
JCD02773@nifty.ne.jp

2017年2月23日

目次

1	幾何学と準備	2
2	作業でシンボルマークを描く	6
3	シンボルマークを対称群を用いて描く	9
4	江戸小紋風に	15
付録 A	J のアドオンとスクリプト	21

はじめに

組市松模様のオリンピックとパラリンピックのマークを交互に並べ、北斎の「新型小紋帳」風の江戸小紋に仕立て上げることが最終目標である。

オリンピック招致委員会のシンボルマークは「女子美」の学生がデザインした。本競技の方は曲折の末に野老朝雄氏の「組市松紋」模様に着いた。氏は建築アート出身で、幾何学を駆使した優れたデザインである。



2種類の幾何学模様のデザインは3種類45個のピースを幾何学模様にならべたもので、オ

リンピックの方は 15 個ずつ 3 の回転群 (P3) で、パラリンピックは鏡映群 (Pm) で構成されている。これをツールではなくすべて幾何学で最初から一個ずつ描いてみよう。

1 幾何学と準備

1.1 C.Reiter の dwin

このグラフィックスを描くのに C.Reiter の *Fractal Visualization and J 4th edition* で用いられたグラフィックス環境 *fvj4* を用いている。J のオリジナルの 2D グラフィックス *gl2* に重ねて使用するもので、J の *addon* に入っている。本格的な導入は *Appendix* を参照

この一つのファイル (*hokusai_olympic0.ijs*) で全て完結するよう *dwin* の必要部分はコピーしておいた。

*1

最初にオブジェクトファイルのロードとセット。

coinsert はオブジェクトが混線しないように別棟に格納するする。

*2

```
require 'gl2 trig '  
coinsert 'jgl2'
```

1.2 最初は 3 種のピースから

最初の 3 枚の図は WEB 上のを *GIMP* という *LINUX* 系のグラフィックツールを用いてマークの部分を取り抜いたものである。

*3

GIMP で取り抜いた画像をグラフ用紙にプリントして、座標を読み取った。

最初は 3 の四角形の要素のサイズと使用個数

Type0	9
Type1	18
Type2	18

*1 この部分の著作権はフリーではない

*2 ロケールを活用している

*3 招致委員会の綺麗な図は全て曲線で描かれ、個々のパターンも微妙に異なるので、再現にはスプラインやベジエ曲線を駆使することとなる。自動でパラメーターを取得するツールがないと再現には手数がかかる

```
NB. -----Olympic datablock
```

```
NB. define 3 rectangle
```

```
RECT0=: 0 0,0 13,13 13,:13 0
```

```
RECT1=: 0 0,0 15,9 15,:9 0
```

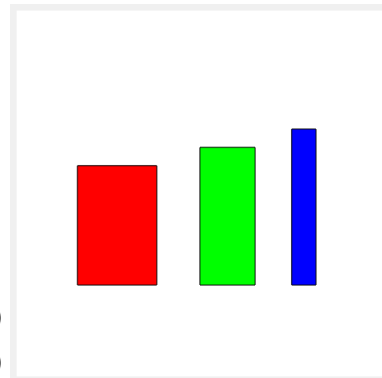
```
RECT2=: 0 0,0 17,4 17,:4 0
```

```
_10 _10 50 30 dwin ''
```

```
255 0 0 dpoly RECT0
```

```
0 255 0 dpoly }:"1(RECT1,.1) mp transm 20 0
```

```
0 0 255 dpoly }:"1(RECT2,.1) mp transm 35 0
```



*4

1.3 準備運動-Homogeneous Coordinates (同時変換/斉時変換)

2D グラフィックス上での拡大縮小 (*elongation*), 回転 (*rotation*), 移動 (*transformation*) を行うパラメーターを合成して一つのマトリクスを生成する。2D では 3×3 の一つのマトリクスを生成する。これを目的の図形に作用させて描く。

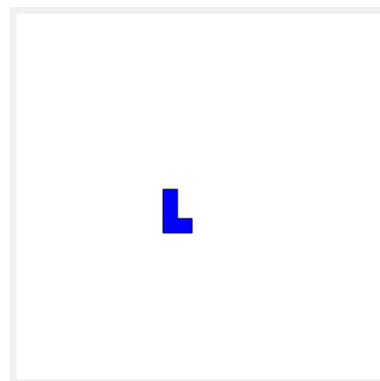
- ベースの図形

```
R2=: 0 0,2 0,2 1,1 1,1 3,:0 3
```

```
R2  
0 0  
2 0  
2 1  
1 1  
1 3  
0 3
```

```
_10 _10 15 15 dwin ''
```

```
0 0 255 dpoly R2
```



GRAMMAR: 最後の連結はラミネート (,:) を用いる

- *elong* 拡大縮小

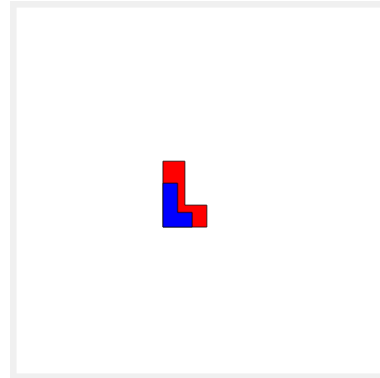
*4 パラリンピックのブロックは印刷サイズで微妙な差異が出たので再計測した

elongm=:3 : '(y,1)* =i.3'

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

```
elongm 1.5 1.5
1.5 0 0
0 1.5 0
0 0 1
```

```
}: "1 (R2,.1) mp elongm 1.5 1.5
0 0
3 0
3 1.5
1.5 1.5
1.5 4.5
0 4.5
```



• rotate 回転

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

```
rotm 3r12p1
0.707107 0.707107 0
-0.707107 0.707107 0
0 0 1
```

rotm=: (cos, sin,0:),(-@sin,cos,0:),: 0: ,0:,1:

回転角はラジアンである。1 ラジアンは 180°

GRAMMAR:

mp 内積 mp=: +/ . *

0: 1: 常に 0,1 を表示する動詞

分数と回転角 J では r と p で記述できる。 $4r12p1 = 1r3p1 = \frac{1}{3}\pi = 60^\circ$

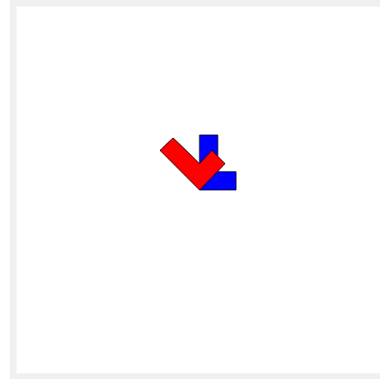
角度はデフォルトはラジアン

```
}:"1 (R2,.1) mp rotm 3r12p1
```

```

      0      0
    1.41421 1.41421
0.707107 2.12132
      0 1.41421
-1.41421 2.82843
-2.12132 2.12132

```



GRAMMAR

R2,.1 縦連結 3×3 のマトリクスに対応するため R2 の右にすべて 1 の列を付加する

}:"1 付加した後尾の列を削除する。"1 はランク指定で列に削除する関数 (}:) を列に作用させる

=i.2 =は等しいかどうかの判定 (代入ではない) 0,1 と 0,1 を判定している。

=i.2 NB. 単位行列を生成するイディオム

```

1 0
0 1

```

- transform 移動

```
transm=: 3 : '(=i.2), y,1'
```

$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix}$$

```

transm 2 3
1 0 0
0 1 0
2 3 1

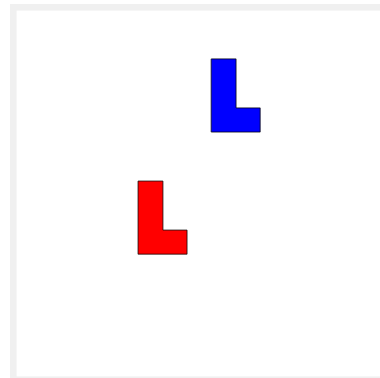
```

```
(<R2),.<}::"1(R2,.1) mp transm 3 5
```

```

+---+---+
|0 0|3 5|
|2 0|5 5|
|2 1|5 6|
|1 1|4 6|
|1 3|4 8|
|0 3|3 8|
+---+---+

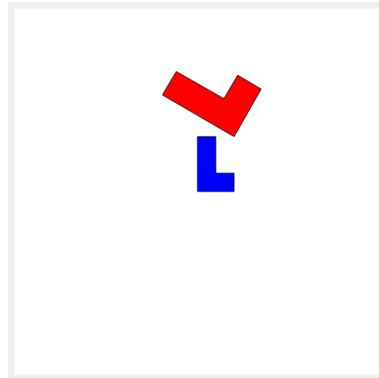
```



- Homogeneous Coordinates 斉次変換

拡大・縮小、回転、移動を同時に行うマトリクスを生成する

```
(elongm 1.5 1.5) mp (rotm 4r12p1) mp transm 2 3
  0.75 1.29904 0
-1.29904  0.75 0
      2      3 1
(R2,.1) mp (elongm 1.5 1.5)
mp (rotm 4r12p1) mp transm 2 3
      2      3 1
      3.5 5.59808 1
      2.20096 6.34808 1
      1.45096 5.04904 1
     -1.14711 6.54904 1
     -1.89711  5.25 1
```



2 力業でシンボルマークを描く

野老氏は建築系なので円に沿って3種計45個の四角形で市松模様を製図版と角度も計測できる定規で精密に組み上げられている。

グラフィックソフトで組み上げたのか、或いは製図版上に精密に角度を割り付けた切り絵を張ったかもしれない。

2.1 図形データ

1. オリンピック、パラリンピックの双方のマークは何れも3種類計45個 (*Type0* 9個、*Type1, Type2* それぞれ18個)の四角形で構成されている
2. 描画は次の2のパラメータをマークを印刷されたグラフ用紙上で読み取る。
 - 基点
四角形の左下(左足)を基点とする。(統一)
 - 角度
角度は全て15°の倍数で構成されている。分度器を求めて確認した。
15度は $\pm 1r12p1$ ラジアン ($\frac{1}{12}\pi$) の倍数で *rotm* に渡す。
3. データは各ピース毎にボックスで囲んで独立してループなしで同時計算し、ポリゴンで同時描画する。

```

, . { RC0
+-----+
| 74 14 _0.523599 |
+-----+
| 108 49 _0.523599|
+-----+
| 108 84 _0.523599|
+-----+
NB. position and angle
NB. Rect0 x9
RC0=:74 14 _1r6p1,108 49 _1r6p1,:108 84 _1r6p1 | 117 104 0 |
RC0=:RC0,117 104 0,68 117 0,:38 99 0
RC0=:RC0,18 91 1r6p1,32 43 1r6p1,:62 25 1r6p1 | 68 117 0 |
読み取り単位は 1mm である (細部では更に微調整になければならない) | 38 99 0 |
+-----+
| 18 91 0.523599 |
+-----+
| 32 43 0.523599 |
+-----+
| 62 25 0.523599 |
+-----+

```

4. *elongm* のパラメータは機種での相違もあるので適宜調整する
5. 外周の円は用いてない
6. パラリンピックのグラフ図形とは切り抜きや印刷の関係で微妙な差異が出た。このため 3 個の四角形のサイズを微調整した。

GRAMMAR

ボックス *J* 独自の機能であり、ネスティッドアレー (二重配列) の簡易版として機能する。
 同時描画 ボックスを用いないと *penup,pendown* しなければならない

2.2 Script

45 個のポリゴンを描く *Script* は次の簡潔なものである。

1. 基点と角度の情報をもとに、四角形を基点ごとに傾けて生成する。

```

move1=: 4 : 0
NB. Usage: RECT0 move1 1.2 1.5 ;1r4p1
'x0 y0 rot'=: y

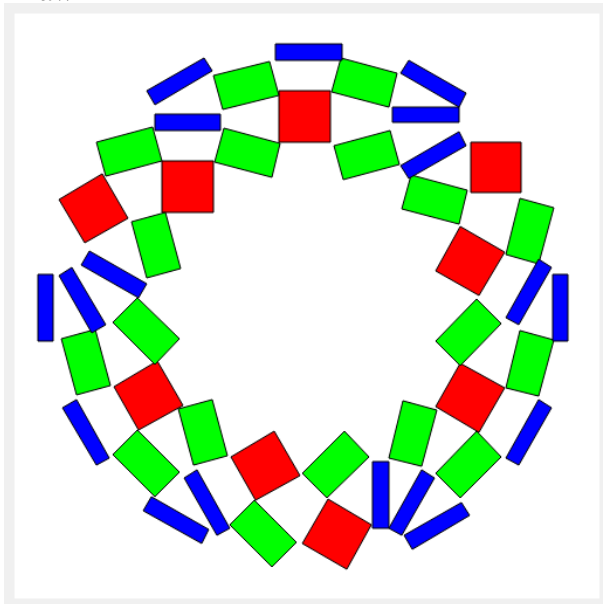
```

```
}: "1(x,.1) mp (rotm rot) mp transm x0,y0  
)
```

2. 描画はこれで足りる

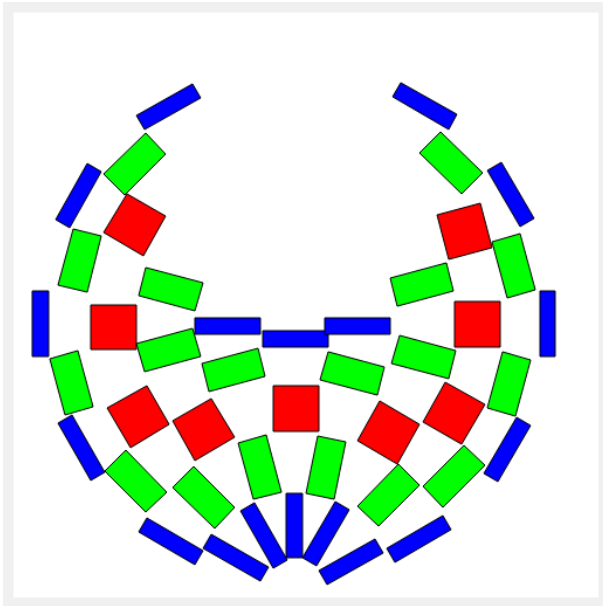
```
draw_olympic=: 3 : 0  
NB. draw_olympic ''  
0 0 150 150 dwin ''  
255 0 0 dpoly L:0 RECT0 move1 L:0 { RC0  
0 255 0 dpoly L:0 RECT1 move1 L:0 { RC1  
0 0 255 dpoly L:0 RECT2 move1 L:0 { RC2  
)
```

3. Draw



2.3 パラリンピック

オリンピックと同様にプリント図形から 45 個の基点と角度を読み取る。基礎になる 3 個のピースのサイズはづ系を見ながら微修正した。



```

draw_paralympic=: 3 : 0
NB. draw_olympic ''
0 0 180 180 dwin ''
255 0 0 dpoly L:0 RECTP0 move1 L:0 { PC0
0 255 0 dpoly L:0 RECTP1 move1 L:0 { PC1
0 0 255 dpoly L:0 RECTP2 move1 L:0 { PC2
)

```

3 シンボルマークを対称群を用いて描く

幾何の世界に 17 種類の対称群がある。「美の幾何学」で中村儀作氏の丁寧な解説がある。
*5

パラリンピックマークは鏡映 (Pm) を使い、オリンピックマークは回転対称 ($P3$) をアレンジしたものをを用いている。

3.1 鏡映 (Pm)

1. トレーニング

- 全体 (x,y) にマイナスを作用させると

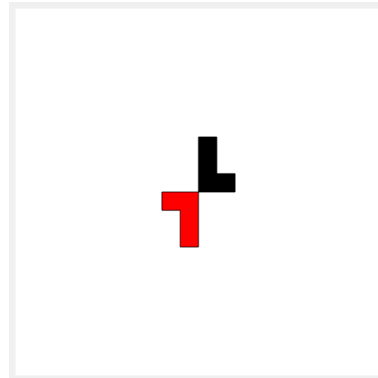
*5 対称群 17 種は P を用いた記号であらわされる。

```

    _10 _10 10 10 dwin ''
    dpoly R2
    255 0 0 dpoly - R2

```

2回の鏡映でも、回転でも生成できる。

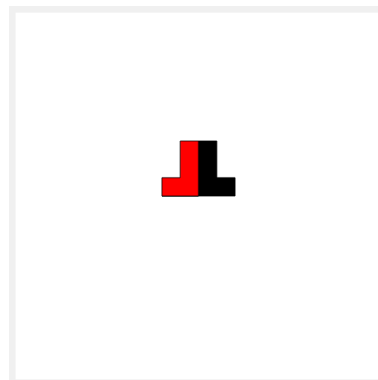


- x 軸のみにマイナスを作用させると (Pm)

```

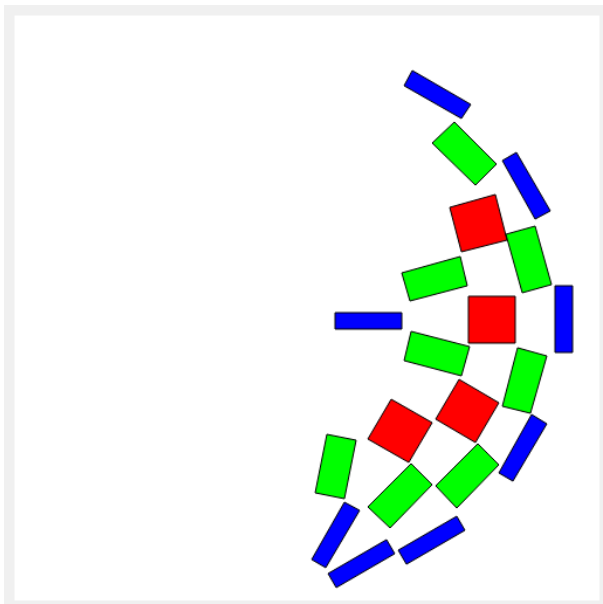
    _10 _10 10 10 dwin ''
    dpoly R2
    255 0 0 dpoly R2 *("1) _1 1

```



2. Step1

中心の3個を除き、右側のピースを取り出す



3. Step2

- 四角形のデータを全て展開する
- 鏡を置く X 軸の点を 0 として、展開したデータを変換する

- X 軸上でデータを反転させる (-)

```
mirror=: 4 : 0
NB. 87 mirror RECTP0;PC10
NB. x is centrer of x coordinate
'RECT POINT'=. y
POINT=. POINT -("1) x,0 ,0
tmp=. RECT move1 L:0 { POINT
tmp ,_1 1 * ("1) L:0 tmp
)
```

4. Step3

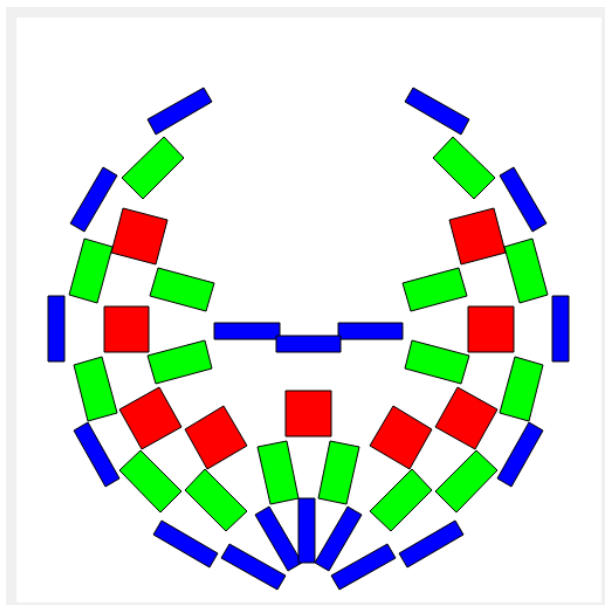
鏡をまたぐ3個の四角形はx軸の中心を0に変換し、別に描く

```
centerpos=: 4 : 'y -("1) L:0 x,0, 0'
```

5. 鏡映での留意事項

傾いた長方形を鏡に映すとき、基点(軸足)が反転する。今回は左下(左足)を基点にして長方形を展開しているが、鏡映では右足に基点が移ってしまう。ダンスなら足が纏れる状態で、描いてわかった

- データを全部展開してから、鏡映に写す
- 図の中心線(鏡に位置)のx座標が0になるようにデータを変換する



6. 描画の SCRIPT

```

draw_paralympic_mirror=: 3 : 0
NB. so many(9) parameter , write each
_90 0 90 180 dwin ''
255 0 0 dpoly (L:0) 87 mirror RECTP0;PC10
0 255 0 dpoly (L:0) 87 mirror RECTP1;PC11
0 0 255 dpoly (L:0) 87 mirror RECTP2;PC12
NB. -----
255 0 0 dpoly L:0 RECTP0 move1 L:0 { 87 centerpos PMC0
0 0 255 dpoly L:0 RECTP2 move1 L:0 { 87 centerpos PMC2
)

```

3.2 回転対称 P3

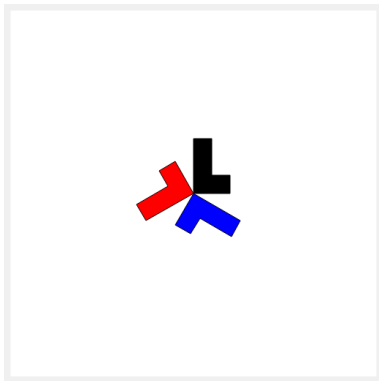
1. トレーニング

- 中心点で回す場合 P3 では回転角は $\frac{2}{3}\pi = 120^\circ$

```

_10 _10 10 10 dwin ''
dpoly R2
255 0 0 dpoly }:"1 (R2,.1) mp rotm 8r12p1
0 0 255 dpoly }:"1 (R2,.1) mp rotm 16r12p1

```



2. STEP1

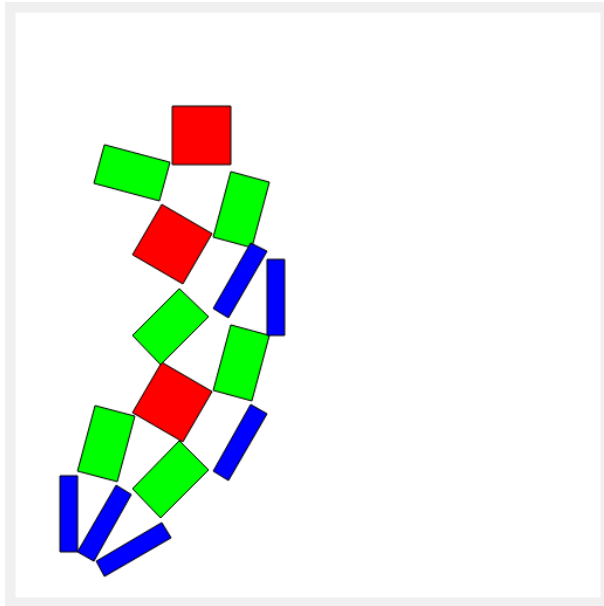
- 15 個のピースの群を取り出す。3 本並んだ短冊の左側のピースの左足を 0 に変換し、15 ピースの座標を再計算する

```

NB. P3 for Olympic
RCR0=: ((>:i.3){RC0) -("1) 92 18 0
RCR1=: ((>:i.6){RC1) -("1) 92 18 0
RCR2=: ((i.6){RC2) - ("1)92 18 0

```

- 描く



3. STEP2

回転させる。回転角は $120^\circ, 240^\circ$ でラジアンでは $\frac{2}{3}\pi, \frac{4}{3}\pi$ となる。

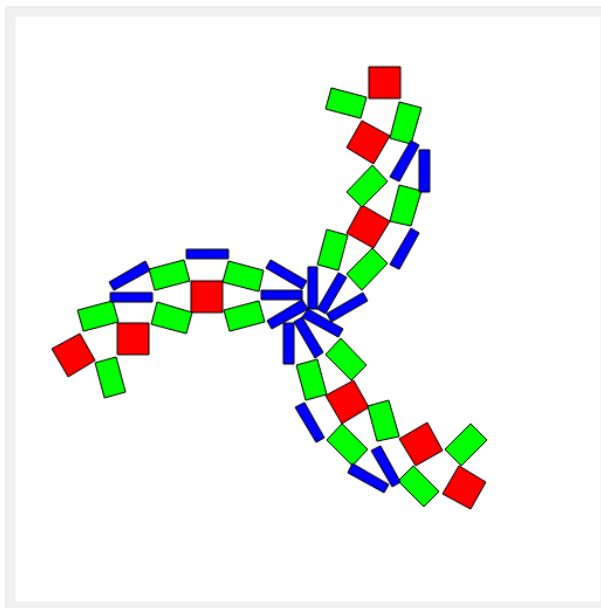
- 基点変換

- この一群の左下のピースの座標は次である。

{. RC2

92 18 0

- ここを 0 に変換して回転の中心として ($No:0$ の基点) 2つの回転角で回転させると **P3** になり、3枚羽根の扇風機になる



4. STEP3

回転でマークを描くには3のブロック(群)毎に、基点(または差分)と回転角を与える。

(a) 図を見ると、RECT0のうち群に取り込んだ3個のブロックの一番上の左上(肩)が次の基点になる。No1の点である。

```
RECT0 move1 L:0 { RCR0
```

```
+-----+-----+-----+
|      16      31|      16      66|25 86|
|  22.5 42.2583|  22.5 77.2583|25 99|  NB. * a little difference
|33.7583 35.7583|33.7583 70.7583|38 99|
|27.2583   24.5|27.2583   59.5|38 86|
+-----+-----+-----+
          Block1      Block2          *Block3          2
```

(b) No2は次により求まる。

No2は0に基点を移してから回転させているので、採寸したグラフ用紙の座標とは異なる。

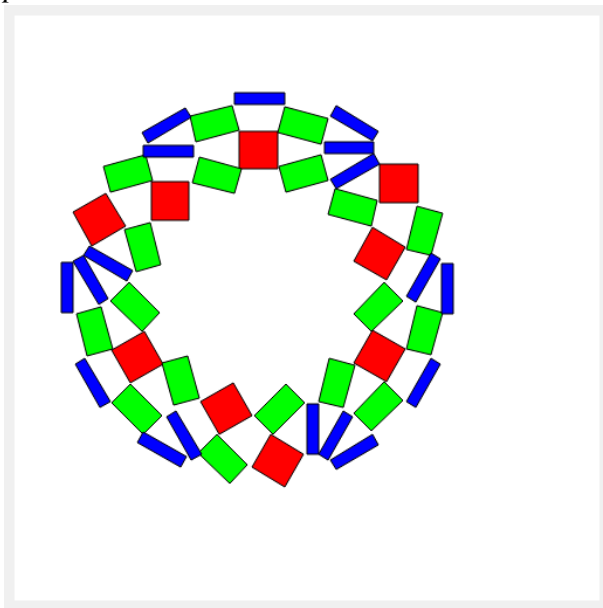
```
tmp3=: 25 99 trans_p3 2r3p1 rot_p3 tmp0
```

(c) No2に移動して回転を与える

```
tmp6=:_74.23 71.15 trans_p3 4r3p1 rot_p3 tmp0
```

5. STEP 4

p3で描く



計算部分のスク립トを示す。経過を確認するため手順を丁寧に書いてある。

```

draw_olympic_p3=: 3 : 0
NB. draw_olympic_p3 '' NB. OK
_100 _50 100 150 dwin ''
NB. 1/3 base
tmp0=: RECT0 move1 L:0 { RCR0
tmp1=. RECT1 move1 L:0 { RCR1
tmp2=. RECT2 move1 L:0 { RCR2
NB. -----
tmp3=: 25 99 trans_p3 2r3p1 rot_p3 tmp0
tmp4=: 25 99 trans_p3 2r3p1 rot_p3 tmp1
tmp5=: 25 99 trans_p3 2r3p1 rot_p3 tmp2
NB. -----
tmp6=:_74.23 71.15 trans_p3 4r3p1 rot_p3 tmp0
tmp7=:_74.23 71.15 trans_p3 4r3p1 rot_p3 tmp1
tmp8=:_74.23 71.15 trans_p3 4r3p1 rot_p3 tmp2
)

```

4 江戸小紋風に

4.1 手法の検討

江戸小紋はそれほどピースが多くないので、描く時に展開して、*dpoly* などに渡しているが、この組市松模様はピースが 45 あるので北斎流の手法のスク립トに載せてさっと描くことはできない。

- 北斎の江戸小紋を描く関数群を活用する
- 2つのマークを単体で描いたものは一辺 120mm ほどあり小紋サイズではないので、縮小パラメーターを組み込む。定数が決まればそれほど変更するものではないので、外つけとする

move0 の中で *elongm 0.2 0.2* と指定すればで 5 分の 1 に縮小できる。オリンピックとパラリンピックのグラフ用紙に印刷した拡大率に微妙に差があり、出来上がった図の大きさを *elongm* で調整する

```

move0=: 4 : 0
NB. Usage: RECT0 move0 L:0 {RC0, .0.2
'x0 y0 rot elong'=: y
}: "1 (x,.1) mp (rotm rot) mp (transm x0,y0) mp elong 2 # elong
)

```

- 小紋は単色ではべたっとした感じになるので、3種の4角形45ピースを色を変化させるため、3回に分けて計算し、表示する。

```
calc_olympic=: 4 : 'x move0 L:0 y '
```

```
NB. Usage: RECT0 move0 L:0 { RC0,.0.2
```

1. 小紋のマトリクス数を指定すると位置決めして表示する

`mk_diff_sub0` はそのまま使える。

```
2 3 mk_diff_sub0 OLPARAM
```

```
+-----+-----+-----+
|0 30|30 30|60 30|
+-----+-----+-----+
|0 0 |30 0 |60 0 |
+-----+-----+-----+
```

2. 北斎の江戸小紋はピースが少ないので分割して色の変化を求めたが、3分割しても9か18のマトリクスと小紋の数だけ位置計算を行わなければならない。ボックスとボックスのマトリクスの相互計算はきわめて複雑である。片方のボックスを開いて3元マトリクスにして、ランク操作を行うこととなるが、素直な解決策が見当たらなかった。
3. 小紋の数と位置をあらわすマトリクスを開いて、ループで計算するスクリプトを作成する

```
mat2plus=: 4 : 0
```

```
NB. SMAT mat2plus tmp2
```

```
mat0=: >, x
```

```
tmp=.<''
```

```
for_ctr. i. # mat0 do.
```

```
tmp0=. (ctr { mat0) +"1 L:0 y
```

```
tmp=.tmp,tmp0
```

```
end.
```

```
} .tmp
```

```
)
```

4. 小紋の各模様位置決め(差分)

```
OLPARAM=: 0 0;30 0;0 30 NB. base ; x diff ; y diff
```

5. x, y の最大値と最小値を求めて、キャンバスの大きさを自動指定する。

```
minmax_dwin=: 3 : ' ((<.@(<./)), >.@(>./)) ; y'
```

6. 小紋での `draw` の手順を簡単にまとめる

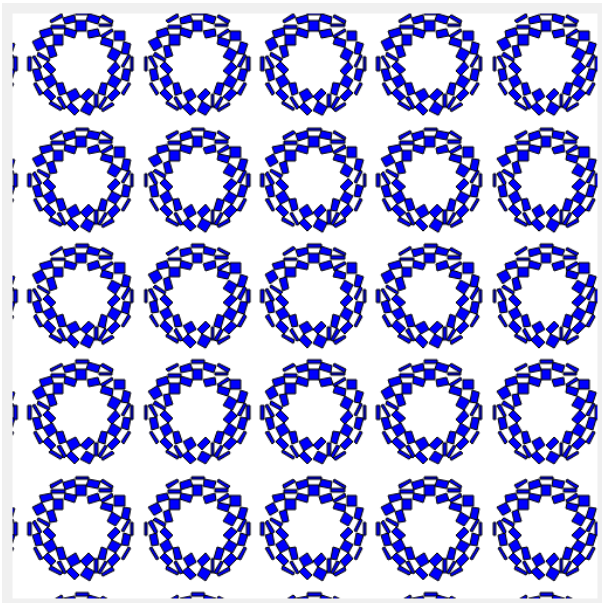

```

    hokusai_olympic  =: 3 : 0
NB. Usage:  hokusai_olympic 5 5
SMAT=. y mk_diff_sub0 OLPARAM
tmp0=. SMAT mat2plus RECT0 calc_olympic {RC0,.0.2
tmp1=. SMAT mat2plus RECT1 calc_olympic {RC1,.0.2
tmp2=: SMAT mat2plus RECT2 calc_olympic {RC2,.0.2
(minmax_dwin tmp2) dwin ''
0 0 255 dpoly L:0 tmp0
0 0 225 dpoly L:0 tmp1
0 0 195 dpoly L:0 tmp2
)

```

4.2 江戸小紋でオリンピックマークを描く

```
hokusai_olympic 7 7
```



4.3 江戸小紋でパラリンピックマークを

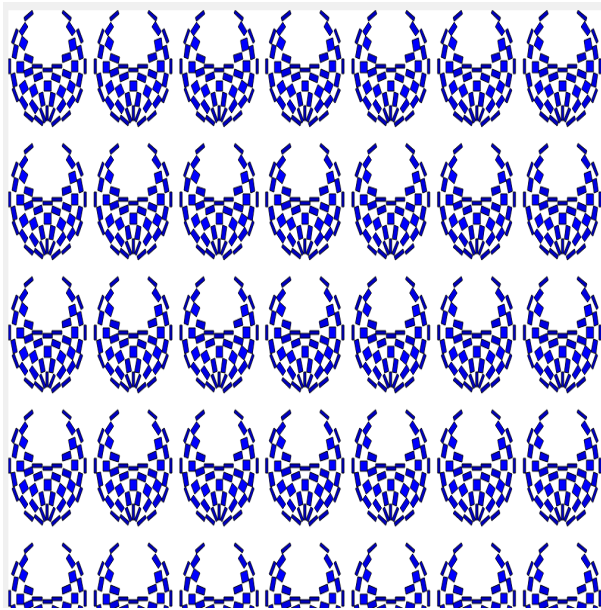
パラリンピックも描いてみよう

一個のサイズが多少異なる。*move0/elongm* のパラメータで調整し、小紋を組み上げる

```

hokusai_paralympic  =: 3 : 0
NB. Usage: hokusai_olympic 5 5
SMAT=. y mk_diff_sub0 OLPARAM
tmp0=. SMAT mat2plus RECTP0 calc_olympic {PC0,.0.17
tmp1=. SMAT mat2plus RECTP1 calc_olympic {PC1,.0.17
tmp2=. SMAT mat2plus RECTP2 calc_olympic {PC2,.0.17
(minmax_dwin tmp2) dwin ''
0 0 255 dpoly L:0 tmp0
0 0 225 dpoly L:0 tmp1
0 0 195 dpoly L:0 tmp2
)

```



4.4 マークをミックスする

小紋でオリンピックとパラリンピックを交互に組合わせる。北斎のスク립トから `index_separate` を流用し、先に用いた `mk_diff_sub0` と組み合わせる

1. 分離するインデックスを作る

```

index_separate 3 3
+-----+-----+
|0 2 4 6 8|1 3 5 7|
+-----+-----+
i.3 3
0 1 2
3 4 5
6 7 8

```

```

3 3 mk_diff_sub0 OLPARAM
+-----+-----+
|0 60|30 60|60 60|
+-----+-----+
|0 30|30 30|60 30|
+-----+-----+
|0 0 |30 0 |60 0 |
+-----+-----+

```

2. インデックスによりポジションデータを分離するスクリプトを作成する。

```

> pick_separate 3 3
+-----+-----+-----+-----+-----+
|0 60 |60 60|30 30|0 0 |60 0|
+-----+-----+-----+-----+-----+
|30 60|0 30 |60 30|30 0|   |
+-----+-----+-----+-----+-----+

```

```

pick_separate=: 3 : 0
NB. usage: pick_separate 4 5
Size=.y
'indOL indPARA'=:index_separate Size
tmp=:Size mk_diff_sub0 OLPARAM
(<indOL { ,tmp),<indPARA{,tmp
)

```

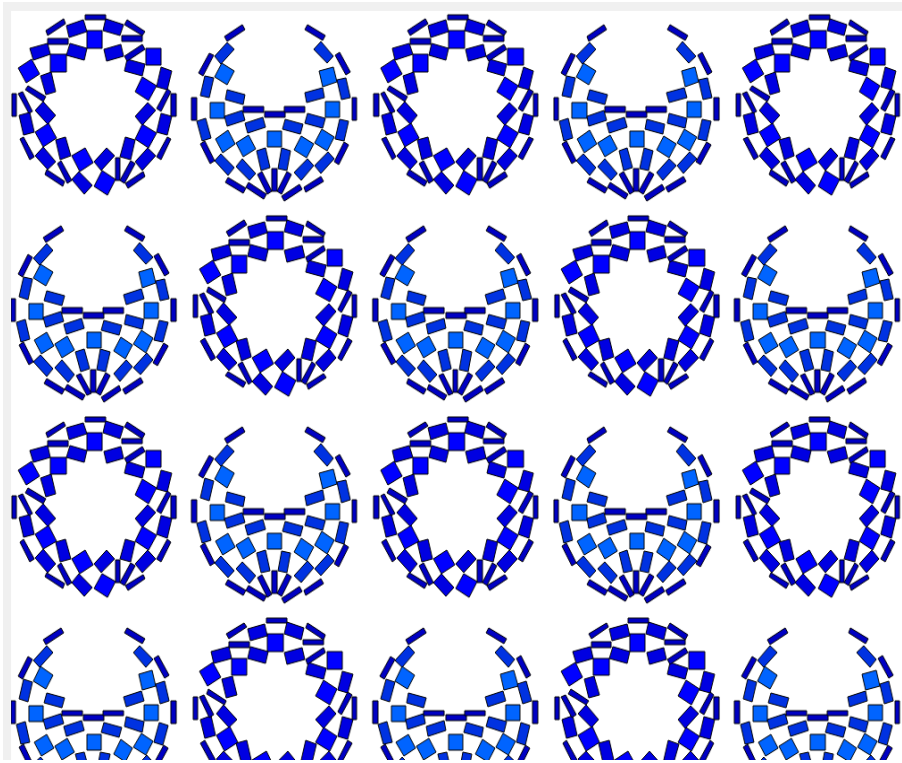
3. 北斎オリンピック江戸小紋のスクリプトを多少改良すればオリンピックとパラリンピックが交互に出てくる小紋が作成できる。次の3行が肝要である

```

'OLMAT PARAMAT'=: pick_separate Size
tmp0=: OLMAT mat2plus RECT0 calc_olympic {RC0,.0.2
tmp3=. PARAMAT mat2plus RECTP0 calc_olympic {PC0,.0.17

```

4. カラーは *RGB* を指定することで自由に変えられる



References

Clifford A. Reiter [*Fractal Visualization and J 3rd.edition*] Lulu.com 2007

伏見康治・安野光雅・中村義作「美の幾何学」早川文庫 2010

付録 A J のアドオンとスクリプト

- J 言語は次から入手できる。

<http://www.jsoftware.com>

`download` → `instalation` をクリックして辿っていくと *DL* ページになる。

WIN/32,64 MAC Linux Raspberrypi(linux) 版がある。最新は *J805* フロントエンドに *QT* を用いた版と *HTML* 版がある。(J8 からの *QT* 版の方が簡単である)

- J の多彩なパッケージは *addon* で提供される。J8 ではネットに繋いで *Tools Package-Manager* を呼び出し、チェックを入れるとインストールされる。大した量ではないので私は全部を導入している。
- このドキュメントでは *graphics/fvj4* を用いている
`addons` → `graphics` → `fvj4` → `dwin` を読み込む。
スクリプトを開いた場合は、`Ctrl+Shift+E` で読み込める
- このスクリプトは次から入手できる。

japla.sakura.ne.jp

workshop 2017/03