J-OpenGLによるルービック・キューブの３Ｄグラフィックス－２
－群論（置換、巡回置換）によるルービック操作とプログラム－

西川　利男

J-OpenGLの3Dグラフィックスを用いたルービック・キューブの動作シミュレーションのプログラムについて、先に報告した [1] 。このときはルービックの動きをJで直接コーディングした。しかし、その後、ルービック・キューブの動作は群論のかっこうのテーマであることが判り、群論でJを使うため何回かの準備を行った。[2]

また、昨年暮に偶然、まさにルービック・キューブのための群論入門書であるD.Joyner,"Adventures in Group Theory: Rubik's Cube" の訳書[3]を見つけた。

今回はその後の進展をもとに、ルービック・キューブを群論－置換群としての見方で、先のJプログラムを見直し、改良を加えたので、それについて述べる。

１．群論とルービック・キューブ

群論とは（数学そのものがそうだが）、図形や構造の運動を考察するのに、同じ挙動を示す(数学的には同形写像 isomorphismと言う）数式の演算処理を利用する考え方のツールだと言えよう。つまり解析幾何と同じに数式の演算のほうが楽だからである。ルービック・キューブに限らず立体幾何の問題はユークリッドやピタゴラスの天才には易しくても、ふつうの頭には至難のわざである。しかし、現代のコンピュータを使った演算ならわれわれにも可能である。群論はそういう道具なのではないだろうか。

２．ルービック・キューブの記述法

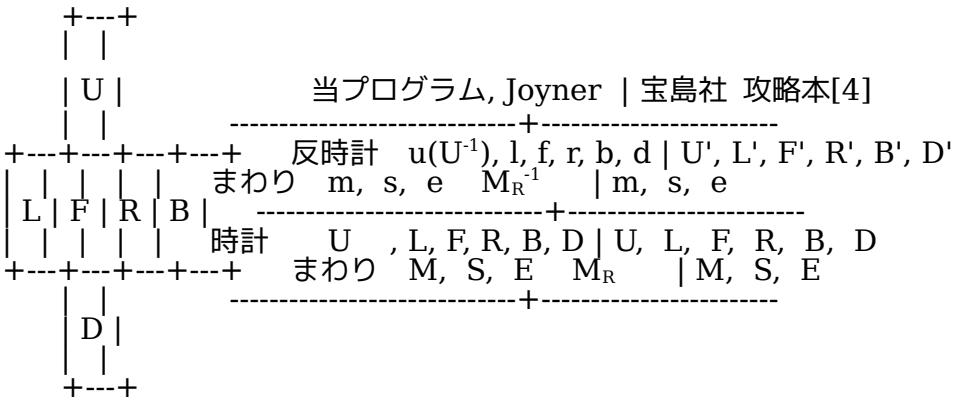
Joyner[3]や最近の攻略本[4]に採用されている以下の国際的記法に従った。前回の島内記法との対応はかっこ内に示す。

上面: U(T)　左面: L(W)　前面: F(S)　右面: R(E)　背面: B(N)　下面: D(B)

そして、ルービックの操作を前回と同様に、これらの面に向かって、正、逆の回転の運動として、面に対応させて名付ける。これらの操作はJでいう動詞である。

本稿では Joyner の本に従い、例えば、上面での反時計まわりの回転を u、時計まわりの回転を U、… とする。「宝島社」の攻略本では U, U' のようにしている。

また、中間の辺を通る回転は U→F→D→B では m(R 面から見て反時計), M(R 面から見て時計)と、U→R→D→B では s, S と、L→F→R→B では e, E とそれぞれ名付けた。



３．置換群（巡回置換と直接置換）によるルービック操作

まず、ルービック面小片(facet)へのインデックスを次のように名付ける。Joyner の本を参考にしたが、われわれは J に合わせて 0-オリジンとし、かつ改良を加えた。

```
    RUBIK_DISPLAY ''
            +--+--+--+
            | 0| 1| 2|
            +--+--+--+
            | 3| 4| 5|
            +--+--+--+
            | 6| 7| 8|
            +--+--+--+
+--+--+--++--+--+--++--+--+--++--+--+--+
| 9|10|11||18|19|20||27|28|29||36|37|38|
+--+--+--++--+--+--++--+--+--++--+--+--+
|12|13|14||21|22|23||30|31|32||39|40|41|
+--+--+--++--+--+--++--+--+--++--+--+--+
|15|16|17||24|25|26||33|34|35||42|43|44|
+--+--+--++--+--+--++--+--+--++--+--+--+
            +--+--+--+
            |45|46|47|
            +--+--+--+
            |48|49|50|
            +--+--+--+
            |51|52|53|
            +--+--+--+
```

これを使って、ルービック操作は群論の置換群（巡回置換の組）として記述される。

例えば、前面での回転 f はつぎのようになる。

```
f =: (18 20 26 24);(19 23 25 21);
       (6 27 47 17);(7 30 46 14);(8 33 45 11)
```

ここでルービック面の個々の小片をつぎのように表す。

```
   RUBJ =: 9#'ULFRBD'
   RUBJ
UUUUUUUUULLLLLLLLLFFFFFFFFFRRRRRRRRRBBBBBBBBBDDDDDDDDD
```

これを見やすくするためのツール display を作った。f 以外のすべての回転および display の定義は稿末のプログラム・リストを参照のこと。なお、実際のプログラムでは、回転 f のコマンドとして ff =: f & C. のような動詞とした。

```
    display RUBJ
      +---+
      |UUU|
      |UUU|
      |UUU|
+---+---+---+---+
|LLL|FFF|RRR|BBB|
|LLL|FFF|RRR|BBB|
|LLL|FFF|RRR|BBB|
+---+---+---+---+
      |DDD|
      |DDD|
```

```
   |DDD|
   +---+
```

前面でのルービック回転操作 f は、インデックス f と J のプリミティブ C.を使って巡回置換として演算する。

```
   f C. RUBJ
UUUUUURRRLLULLULLUFFFFFFFFFDRRDRRDRRBBBBBBBBBLLLDDDDDD
   display f C. RUBJ
     +---+
     |UUU|
     |UUU|
     |RRR|
+---+---+---+---+
|LLU|FFF|DRR|BBB|
|LLU|FFF|DRR|BBB|
|LLU|FFF|DRR|BBB|
+---+---+---+---+
     |LLL|
     |DDD|
     |DDD|
     +---+
```

さらに、ルービックの逆回転の操作 F は J の逆演算(^:_1)により行える。

```
   display f C.^:_1 RUBJ
     +---+
     |UUU|
     |UUU|
     |LLL|
+---+---+---+---+
|LLD|FFF|URR|BBB|
|LLD|FFF|URR|BBB|
|LLD|FFF|URR|BBB|
+---+---+---+---+
     |RRR|
     |DDD|
     |DDD|
     +---+
```

このようにして、群論の巡回置換の演算によりルービック操作が行われた。

ところで、今の操作は直接置換ではどうなるだろうか。直接置換のインデックス f_p は次のようになる。

```
   f_p =: , > ff C. i.54
   f_p
0 1 2 3 4 5 27 30 33 9 10 8 12 13 7 15 16 6 20 23 26 19 22 25 18 21 24 47
28 29 46 31 32 45 34 35 36 37 38 39 40 41 42 43 44 11 14 17 48 49 50 51
52 53
   $f_p
54
```

上の結果では分かり難いので、普通の数学表記の直接置換に合わせて(ただし 0 – オリジン)、さらに分割して見やすくする。

```
   20 {."(1) (i.54) ,: f_p
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 27 30 33 9 10  8 12 13  7 15 16  6 20 23

   20 {."(1) 20 }."(1) (i.54) ,: f_p
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
26 19 22 25 18 21 24 47 28 29 46 31 32 45 34 35 36 37 38 39
```

```
    14 {."(1) 40 }."(1) (i.54) ,: f_p
40 41 42 43 44 45 46 47 48 49 50 51 52 53
40 41 42 43 44 11 14 17 48 49 50 51 52 53
```

　またこれにより、直接置換は実用的ではなく、実際には巡回置換で行うことになることがわかる。

なお当然、直接置換のインデックス f_p を用いても、同じルービック操作は行える。

```
   display f_p C. RUBJ
      +---+
      |UUU|
      |UUU|
      |RRR|
+---+---+---+---+
|LLU|FFF|DRR|BBB|
|LLU|FFF|DRR|BBB|
|LLU|FFF|DRR|BBB|
+---+---+---+---+
      |LLL|
      |DDD|
      |DDD|
      +---+
```

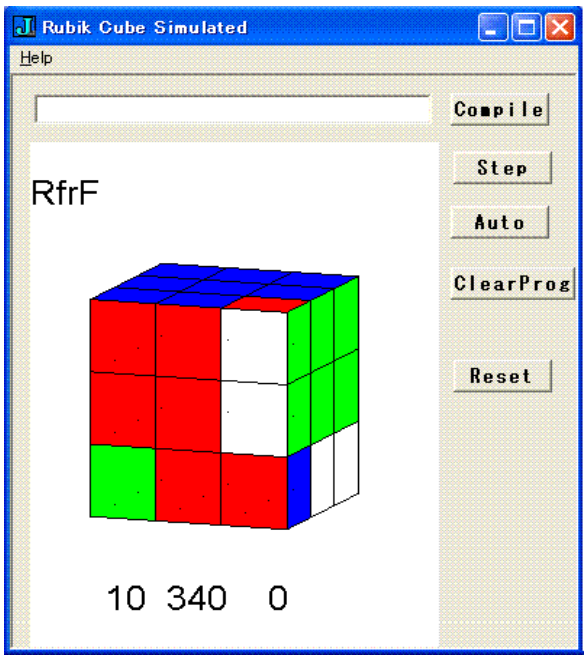
４．J-OpenGL による 3D グラフィックス

ルービック・キューブの J-OpenGL グラフィックスについては先に何回も報告した。いままでとの違いは、ルービック操作を直接、J での配列処理でではなく、群論の操作で行うため、ルービック面小片(facet)を名前の文字で表し、最後に色指定をするようにした。これにより仕様の異なったルービック・キューブに対応できるようになる。

プログラムの全体は先に示したものとほとんど同じであるが、修正した色指定の部分を挙げる。

```
NB. new version 2012/1/17
=========================================
sel_col =: 3 : 0
 KK =: 2#y. NB. KK is Rubik command such as ff, FF, mm, MM, ...
 RUBX =: ". KK,  ' RUBX'  NB. move Rubik facets
 NB. Coloring Facets calling 'col' subprogram
 'T_COL W_COL S_COL E_COL N_COL B_COL' =: <"(1) 6 9$ col RUBX
)
```

以下は F 面の右下隅の facet を U 面の右下隅に移動させる基本ルーチンである。しかし、このとき同時に、その他の多くの facet にも影響を与えている。



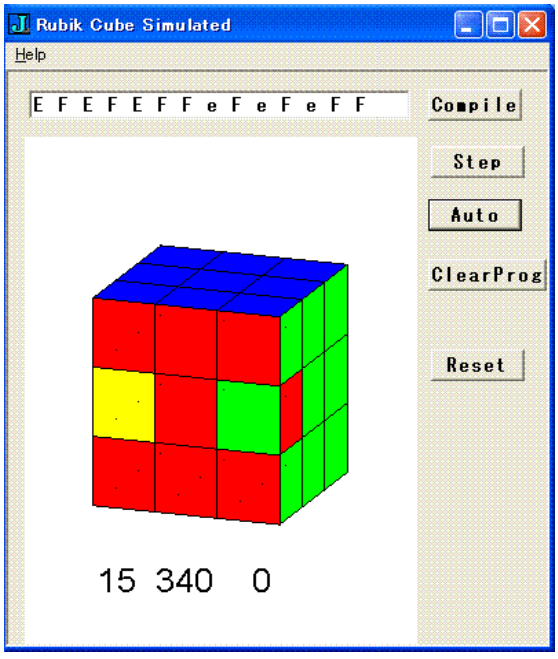
これに対応する display を用いたプリント表示はつぎのようになる。

```
 display (FF@rr@ff@RR) RUBJ
    +---+
    |UUU|
    |UUU|
    |UUF|
+---+---+---+---+
|LLL|FFD|RRR|BBB|
|LLL|FFD|RRR|BBB|
|LLD|RFF|UDD|LBB|
+---+---+---+---+
    |BRR|
    |DDF|
    |DDF|
    +---+
```

ここで ff はルービック回転操作 f(反時計)に、FF は操作 F(時計)に、rr は操作 r(反時計)に、RR は操作 R(時計)にそれぞれ対応する J の動詞である。なお、J の右からの動詞の実行順序に注意、グラフィックスでの逐次実行とは逆になる。

次はルービック・マヌーバと呼ばれているすばらしい攻略法[4]である。他に影響を及ぼさずに 2 箇所だけの facet の向きを合わせるという操作である。

ここでは、入力窓にあらかじめ操作を入れ、Auto ボタンを押すと一瞬で実行される。また、Step ボタンではワンステップずつ行う。その結果は以下のとおりである。



文献

[1] 西川利男「J-OpenGL によるルービック・キューブの３Ｄグラフィックス」
JAPLA 研究会資料 2011/10/22

[2] 西川利男「J 言語からの群論の理解 - その２」JAPLA 研究会資料 2011/11/26

「J 言語からの群論の理解 – その 3 – 直接置換、巡回置換、互換、隣接互換 – 」
JAPLA シンポジウム資料 2011/12/10
[3] D. Joyner, 川辺治之訳「群論の味わい – 置換群で解き明かすルービック・キューブと 15 パズル」共立出版(2010).
[4] 「頭を鍛えるルービックキューブ完全解析！」宝島社(2007).

Jのプログラムリスト

```
NB. OpGLN_RubikJ.ijs  2012/1/6, NJoyF checked and revised 2012/1/23
NB. Rubik Cube Simulation in Joyner's Group Theory Notation (use cycle)
NB. revised 2012/1/17

NB. revised from OpGLN_Rubik.ijs 2011/9/19
NB. referred from OpGLN3.ijs

NB. run '' => 宝島社「頭を鍛える」色設定
NB. run 0  => 島内 色設定
NB. run 1  => 宝島社「頭を鍛える」色設定

NB. 宝島社「頭を鍛える」コーナー・キューブの向き転換
MPat1 =: 'NbbnbNbnbb' NB. p.19
NB. 宝島社「頭を鍛える」エッジ・キューブの位置転換
MPatA =: 'eiEIEie'    NB. p.21 A
MPatB =: 'EIeieIE'    NB. p.21 B
MPatC =: 'EIeeIEI'    NB. p.21 C

MPatR0 =: 'IEIEIE EiEiEi EE' NB.  p.25 Rubik's Maneuver

SPat2 =: 'eWtnnttnnttnntEwNtWtnnttnnttnntwTn' NB. Shimanouchi p.36,
25a

wr =: 1!:2&2
require 'gl3'

AA=: 0 : 0
pc aa closeok;pn "Rubik Cube Simulated";
menupop "&Help";
menu help "&Help" "" "" "";
menupopz;
xywh 6 22 200 200;cc g isigraph ws_clipchildren ws_clipsiblings rightmove
bottommove;
xywh 211 93 34 11;cc Reset button;
xywh 211 25 34 11;cc Step button;
xywh 210 43 34 11;cc Auto button;
xywh 7 6 197 11;cc Program edit ws_border es_autohscroll;
xywh 210 63 43 11;cc ClearProg button;
xywh 210 6 34 11;cc Compile button;
pas 0 0;
rem form end;
)

NB. Colors from Facets of Rubik
col =: 3 : 0"0
select. y.
   case. 'U' do. 'B'
   case. 'L' do. 'Y'
   case. 'F' do. 'R'
   case. 'R' do. 'G'
   case. 'B' do. 'O'
   case. 'D' do. 'W'
end.
)


NB. Color Definition ========================
NB.   revised for OpenGL spec. 2011/10/12
```

```
R =: 1   0   0     NB. red
O =: 1   0.45 0    NB. orange
NB. O =: 1   0.64 0    NB. pale orange
NB. O =: 1   0.27 0    NB. orange red
Y =: 1   1   0     NB. yellow
G =: 0   1   0     NB. green
W =: 1   1   1     NB. white
B =: 0   0   1     NB. blue


run=: aa_run
aa_run=: 3 : 0
wd :: ] 'psel a;pclose'
wd AA
glaRC ''
ROT =: 0 0 0  NB. rename from RR
Key =: ''
glaFont 'arial 30'
NB. glaUseFontBitmaps 0 32 26 32  Only Number Font
glaUseFontBitmaps 0 32 95 32  NB. Use Character Font
NB. Initial Color Setting for Rubik Cube ==========
RUBX =: ,>RUBJ
NJoyF ''
if. 0 = #initC do. initC =: 10 end. NB. デフォルト＝宝島社の色設定
initC =: y.
init_color initC  NB. change color patterns in several books
istep =: 0
wd 'pshow;ptop'
)

init_color =: 3 : 0
  select. y.
    case. 0 do. NB. 島内　色設定
      S_COL =: 'YYYYYYYYY'
      E_COL =: 'OOOOOOOOO'
      W_COL =: 'RRRRRRRRR'
      T_COL =: 'WWWWWWWWW'
      B_COL =: 'BBBBBBBBB'
      N_COL =: 'GGGGGGGGG'
    case. 1  do. NB. 宝島社「頭を鍛える」色設定
      S_COL =: 'RRRRRRRRR'
      E_COL =: 'GGGGGGGGG'
      W_COL =: 'YYYYYYYYY'
      T_COL =: 'BBBBBBBBB'
      B_COL =: 'WWWWWWWWW'
      N_COL =: 'OOOOOOOOO'
    case. 10 do. NB. adjust to Joyners notation
      T_COL =: col 9#'U'
      W_COL =: col 9#'L'
      S_COL =: col 9#'F'
      E_COL =: col 9#'R'
      N_COL =: col 9#'B'
      B_COL =: col 9#'D'
  end.
)
```

```
NB. display the model picture
=========================================
aa_g_paint =: verb define
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT
draw_rubik ''
draw_frames ''
drawtext''
glaSwapBuffers ''
)


NB. key-in x, y, z, X, Y, Z for rotation ================
aa_g_char =: verb define
ROT =: 360 | ROT + 5 * 'xyz' = 0 { sysdata  NB. rename from RR
ROT =: 360 | ROT - 5 * 'XYZ' = 0 { sysdata  NB. rename from RR
NB. Change Color of Cubies for Rubik Moves ==============
KK0 =. 0 { sysdata
KK1 =: KK0 -. 'xyzXYZ'
Key =: Key, KK1 NB. record of moves
sel_col KK1         NB. move Rubik and change color / new
NB. sel_color KK0   NB. move Rubik and change color / old
glpaintx''
)

NB. new version 2012/1/17
=========================================
sel_col =: 3 : 0
  KK =: 2#y. NB. KK is Rubik command such as ff, FF, mm, MM, ...
  RUBX =: ". KK,  ' RUBX'  NB. move Rubik facets
  NB. Coloring Facets
  'T_COL W_COL S_COL E_COL N_COL B_COL' =: <"(1) 6 9$ col RUBX
)

NB. Revised Joyner's Notation by T. Nishikawa 2012/1/3 ===========
NB. 0-origin and use C.

ALL_COJ =: 9#'BYRGOW'  NB. Actual Rubik'S Color Set
RUBJ    =: 9#'ULFRBD'  NB. for Joyner's Facet Name

NB. Modified Joyner's Functions by T. Nishikawa 2012/1/5
===========
NB. NJoyF ''
NB. e.g. display rr ALL_COL

NB. 2012/1/23 checked OK! & revised  2012/1/23
NJoyF =: 3 : 0
r1 =. 27 29 35 33
r2 =. 28 32 34 30
r3 =. 20 2 42 47
r4 =. 23  5 39 50
r5 =. 26  8 36 53
r0  =: r1;r2;r3;r4;r5
rr =: r0 & C.        NB. verb right turn = counter clockwise turn
RR =: rr^:_1         NB. verb RIGHT turn = clockwise turn

f1 =. 18 20 26 24
f2 =. 19 23 25 21
f3 =. 11  8 33 45
```

```
f4 =. 14  7 30 46
f5 =. 17  6 27 47
f0 =: f1;f2;f3;f4;f5
ff =: f0 & C.
FF =: ff^:_1

l1 =. 9 11 17 15     NB. l is el, not one !!
l2 =. 10 14 16 12
l3 =. 6 38 51 24
l4 =. 3 41 48 21
l5 =. 0 44 45 18
l0 =: l1;l2;l3;l4;l5 NB. l0 is el zero
ll =: l0 & C.        NB. ll is el el
LL =: ll^:_1

b1 =. 36 38 44 42
b2 =. 37 41 43 39
b3 =.  2  9 51 35
b4 =.  1 12 52 32
b5 =.  0 15 53 29
b0 =: b1;b2;b3;b4;b5
bb =: b0 & C.       NB. revised 2012/1/23 clockwise turn
BB =: bb^:_1        NB. revised 2012/1/23 counter clockwise turn

u1 =. 0 2 8 6
u2 =. 1 5 7 3
u3 =. 9 36 27 18
u4 =. 10 37 28 19
u5 =. 11 38 29 20
u0 =: u1;u2;u3;u4;u5
uu =: u0 & C.
UU =: uu^:_1

d1 =. 45 47 53 51
d2 =. 46 50 52 48
d3 =. 24 33 42 15
d4 =. 25 34 43 16
d5 =. 26 35 44 17
d0 =: d1;d2;d3;d4;d5
dd =: d0 & C.       NB. revised 2012/1/23 clockwise turn
DD =: dd^:_1        NB. revised 2012/1/23 counter clockwise turn

e1 =. 12 21 30 39
e2 =. 13 22 31 40
e3 =. 14 23 32 41
e0 =: e1;e2;e3
ee =: e0 & C.
EE =: ee^:_1

m1 =. 1 19 46 43
m2 =. 4 22 49 40
m3 =. 7 25 52 37
m0 =: m1;m2;m3
mm =: m0 & C.
MM =: mm^:_1

s1 =. 3 28 50 16
s2 =. 4 31 49 13
s3 =. 5 34 48 10
```

```
s0 =: s1;s2;s3
ss =: s0 & C.
SS =: ss^:_1
''
)


NB. eg. display ALL_COL
display =: 3 : 0  NB. revised from displv3
y =. 6 9$y.
RD0 =. (3, 3)$ L:0 <"(1) }. }: y
RD1 =. (3, 3)$ L:0 (9#' ');({. y);(9#' ');(9#' ')
RD2 =. (3, 3)$ L:0 (9#' ');({: y);(9#' ');(9#' ')
RD3 =. RD1, RD0,: RD2
RD4 =. ": RD3
RD5 =. (' ') (<(i.4);(i.4)) } RD4
RD6 =. (' ') (<(i.4);(9+i.8)) } RD5
RD7 =. (' ') (<(9+i.4);(i.4)) } RD6
RD8 =. (' ') (<(9+i.4);(9+i.8)) } RD7
)


NB. indicate rotated angle values x, y, z in degree ===========
drawtext =: verb define
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glColor 0 0 0 0
glRasterPos_2.5 2.0 0
glCallLists Key
glRasterPos _2 _2.4 0
glCallLists (5 ": ROT)  NB. rename from RR
)

aa_help_button =: verb define
wd 'mb OpenGL *Keys, x/X, y/Y, z/Z rotate, e/E, w/W, t/T, b/B, s/S, n/N move
Rubik. '
wd 'setfocus g'
)

aa_Reset_button=: 3 : 0
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT
init_color initC
draw_rubik ''
draw_frames ''
Key =: ''
drawtext ''
istep =: 0
glaSwapBuffers ''
wd 'set Program ""'
iPDA =: ''
wd 'setfocus g'
)

aa_Program_button=: 3 : 0
PDA =: Program
iPDA =: ''
wd 'setfocus g'
```

)