

## J-OpenGL によるルービック・キューブの3Dグラフィックス

西川 利男

ルービック・キューブの愛好者が最近また増えている。従来からの正統的 3x3 面のものに加えて、2x2 や 4x4 あるいはもっと変形などいろいろなものがある。

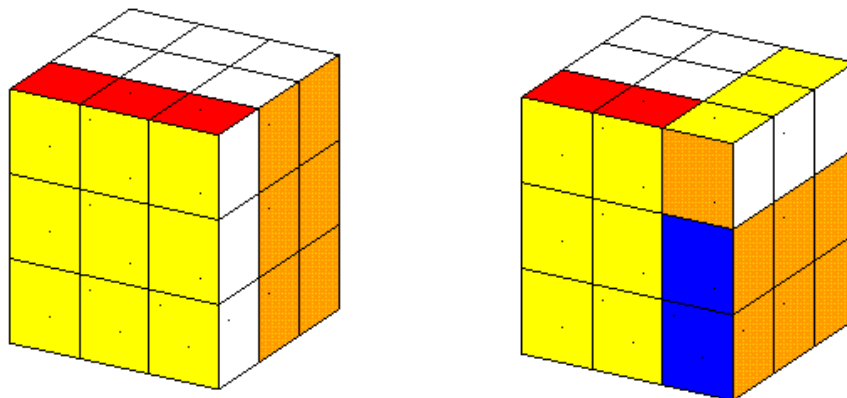
ルービック・キューブが登場したのは1980年代であろうか。若者たちのファッションともいえるブームとなり、一方、島内剛一先生をはじめとして、数学セミナー誌などに紹介やら解法の解説、理論など[1]もたくさんあらわれた。島内先生の本は、また再版されているようである。[2]

私自身も、北海道の中野嘉弘先生と共同で、Jによるルービック・キューブの動きのシミュレーションのプログラムを開発し、紀要として発表した。[3] そのときは立体の展開図として、その動きを表示せざるを得なかった。

今回、J-OpenGLにより動きを伴う3Dグラフィックスが手軽に行えることから、ルービック・キューブの動きをよりリアルに実現することができた。コンピュータ上のシミュレーションを感じさせないほど、実物で動かすのと同じ感覚で楽しむことが出来るのはまさに驚異である。

### 1. ルービック・キューブの各面と動きの記述

ルービック・キューブの各面と動きは紀要のときと同様、島内先生の本[1]に倣った。すなわち、東西南北と上下とした極めてユニークな数学者らしい、そして的確な座標の取り方となっている。



[1] 島内剛一「ルービック・キューブ免許皆伝」日本評論社(1981) など

[2] 瀬山士郎編「別冊日経サイエンス 数学は楽しい Part 2」p.6-13

日経サイエンス社(2010).

I. クリッツ、P. シーゲル「ルービック・キューブを超えて群論パズルを楽しむ」

[3] 西川利男、中野嘉弘、林 雄二

「J言語によるウィンドウズ・プログラミングルービック・キューブのシミュレーション」 北海道情報大学紀要、第10巻第1号、p.219-240 (1998).

ルービック・キューブの最初の状態では次のようになっている。

正面を S(South) 色は Y(yellow), 背面を N(North) 色は G(green)  
右面を E(East) 色は O(orange), 左面を W(West) 色は R(red)  
上面を T(Top) 色は W(white), 底面を B(Base) 色は B(blue)

ルービック・キューブの6つある面はそれぞれ  $3 \times 3 = 9$  個のキュービーと呼ぶ小片から成っていて、ルービック・キューブ操作により独立に変化する。なお、キュービーには3種類の異なるものから成り、つぎのように名付ける。

センター・キュービー = 島内先生の本では1面体と呼んでいる  
エッジ・キュービー = 島内先生の本では2面体と呼んでいる  
コーナー・キュービー = 島内先生の本では3面体と呼んでいる

次に、ルービック・キューブ操作（キュービーの移動）は、面の命名に対応して次のように名付ける。

S(South)を軸とした時計向きの回転を s, 反時計向きの回転を S  
E(East)を軸とした時計向きの回転を e, 反時計向きの回転を E  
W(West)を軸とした時計向きの回転を w, 反時計向きの回転を W  
T(Top)を軸とした時計向きの回転を t, 反時計向きの回転を T  
B(Base)を軸とした時計向きの回転を b, 反時計向きの回転を B  
N(North)を軸とした時計向きの回転を n, 反時計向きの回転を N

ここで大切なことは、上のいずれの操作を行っても、センター・キュービーは移動せず、最初のままに保たれる。島内先生の本では、中通りの回転は、センター・キュービーの移動となるので禁止しているが、便利のため、これも追加した。

S → E → N → W の中通りの回転を i, 反対の回転を I  
S → T → N → B の中通りの回転を j, 反対の回転を J  
E → T → W → N の中通りの回転を k, 反対の回転を K

さてそれでは、最初の状態から実際にルービック操作を行ってみよう。

まず、s の回転をすると、ルービックの面の变化は前ページの左図のようになる。つづいて、e の回転をすると、ルービックの面の变化は前ページの右図のようになる。

ここまですべて J で記述してみる。そのためにはそれぞれのキュービー面を個々に扱わなくてはならないが、

「各キュービーは色の名前の文字をその値として持つ」とする。

最初の状態から s 操作を行った状態（前ページ左図）は

S: YYY    T: WWW    E: WOO    以下略す  
   YYY        WWW        WOO  
   YYY        RRR        WOO

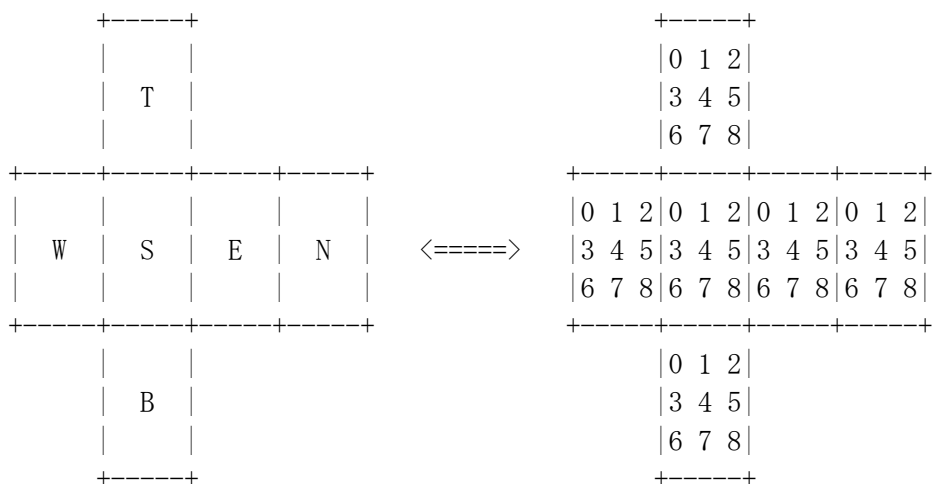
つづいて e 操作を行った状態（前ページ右図）は

S: YYO    T: WWY    E: WWW    以下略す  
   YYB        WWY        000  
   YYB        WWY        000

このようなルービック操作に対する面の变化をプログラミングすることになる。

## 2. Jによるルービック・キューブ操作のプログラム

Jでプログラミングするためには、 $9 \times 6 = 54$ 個のルービック・キューバーをどう表すかが問題である。配列を用いる方法もあるが、ここでは簡単のため直接インデックス指定で行うことにした。その配置を展開図で示すと次のようになる。



この方式を用いて、Jでルービック操作を行う動詞を sel\_color としてプログラミングした。その一部を示す。全体の定義は後のプログラムリストを参照のこと。

```

sel_color =: 3 : 0
S_C =. S_COL
T_C =. T_COL
E_C =. E_COL
W_C =. W_COL
N_C =. N_COL
B_C =. B_COL
select. y.
  case. 's' do.
    S_COL =: , |: |. (3 3)$S_C
    T_COL =: ((8 5 2){W_C} (6 7 8)} T_C
    E_COL =: ((6 7 8){T_C} (0 3 6)} E_C
    B_COL =: ((0 3 6){E_C} (2 1 0)} B_C
    W_COL =: ((2 1 0){B_C} (8 5 2)} W_C
  case. 'e' do.
    E_COL =: , |: |. (3 3)$E_C
    S_COL =: ((2 5 8){B_C} (2 5 8)} S_C
    T_COL =: ((2 5 8){S_C} (2 5 8)} T_C
    N_COL =: ((8 5 2){T_C} (0 3 6)} N_C
    B_COL =: ((0 3 6){N_C} (8 2 5)} B_C
  .....
(途中省略)
.....

```

)

これをテストするための小さな動詞 test\_rubik と test\_initial を定義して、ルービック操作をやってみよう。

```
test_initial =: 3 : 0
  S_COL =: 'YYYYYYYYY'
  E_COL =: '000000000'
  W_COL =: 'RRRRRRRRR'
  T_COL =: 'WWWWWWWWW'
  B_COL =: 'BBBBBBBBB'
  N_COL =: 'GGGGGGGGG'
empty ''
)
```

```
test_rubik =: 3 : 0
R0 =. S_COL;T_COL;W_COL;E_COL;N_COL;B_COL
empty&sel_color y.
R1 =. S_COL;T_COL;W_COL;E_COL;N_COL;B_COL
RR0 =: (3, 3)&$ L:0 R0
RR1 =: (3, 3)&$ L:0 R1
wr ' S T W E N B'
RR0,: RR1
)
```

最初の色の設定を行う。

```
test_initial ''
```

ルービック操作の s をテストしてみる。

```
test_rubik 's'
```

```
S T W E N B
```

```
+---+---+---+---+---+---+
|YYY|WWW|RRR|000|GGG|BBB|
|YYY|WWW|RRR|000|GGG|BBB|
|YYY|WWW|RRR|000|GGG|BBB|
+---+---+---+---+---+
|YYY|WWW|RRB|WOO|GGG|000|
|YYY|WWW|RRB|WOO|GGG|BBB|
|YYY|RRR|RRB|WOO|GGG|BBB|
+---+---+---+---+---+

```

つづいて、ルービック操作の e をテストしてみる。

```
test_rubik 'e'
```

```
S T W E N B
```

```
+---+---+---+---+---+---+
|YYY|WWW|RRB|WOO|GGG|000|
|YYY|WWW|RRB|WOO|GGG|BBB|
|YYY|RRR|RRB|WOO|GGG|BBB|
+---+---+---+---+---+
|YYO|WWY|RRB|WWW|GGG|OOG|
|YYB|WWY|RRB|000|WGG|BBG|
|YYB|RRY|RRB|000|WGG|BBG|

```

+-----+-----+-----+-----+

### 3. 3DグラフィックスためのOpenGLプログラム

JのOpenGLプログラムについては、これまで何度も紹介し、解説してきた。[4]

[4] 西川利男「Jのgl3-OpenGLによるグラフィックスーその1」

JAPLA 研究会資料 2009/9/26 など

JのOpenGLの詳細はそちらを見ていただくとして、ポイントは次のようである。

Jのgl3命令を用いるウィンドウズ・スクリプトを作成する。

J4ではglARCにより直接、J5、J6ではOOP機能によりクラス・インスタンスを作り、J-OpenGLを実行する。

runによるプログラムの起動とともに、以下のグラフィックのチャイルドgのプログラムが自動的に実行される。

g\_paint 画像の頂点、図形などの作成（レンダリング）

g\_char キーボード入力によるイベントの実行

g\_size 画像の視点、投影などの処理

さらにプログラムの構造詳細は次のようになる。

g\_paintの中では、ルービックの図形作成、画像表示など

draw\_rubuk キュービー図形の作成

Point 各面の3 x 3 = 9個のキュービー図形の中心座標

boundary キュービー図形の頂点座標の生成

face\_col キュービー図形を指定した色で塗る

polygon 座標から正方形を作る

draw\_frames キュービー境界線グリッドの作成

face\_frame

fpolygon

g\_charの中では、キーボード入力によるルービック操作と視点座標の回転操作を行う。

sel\_color ルービック操作('s'や'e'など)により色を変える

具体的なJプログラムの詳細は後のプログラムリストを見ていただきたい。ここでは、いくつかの注意すべきプログラム・コーディングを説明する。

OpenGLでは、色の指定は次のように0から1の値で行う。

R =: 1 0 0 NB. red

O =: 1 0.64 0 NB. orange

Y =: 1 1 0 NB. yellow

G =: 0 1 0 NB. green

W =: 1 1 1 NB. white

B =: 0 0 1 NB. blue

ルービック・キューブ操作の最もポイントとなる各キュービーの色の変更は、キーボード入力した's'や'e'などの文字に従い、各面のキュービーを表すインデックスの更新処理である。つまり、先に述べた動詞sel\_colorをバグなくコーディングするかにかかっている。

とくに後ろに隠れて見えない面のインデクシングは最も気を使い、実物の展開図をたどって正確を期すことが、極めて大切であることを痛感した。

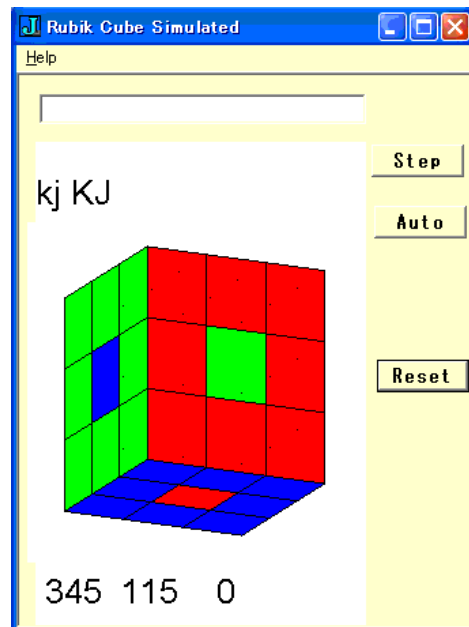
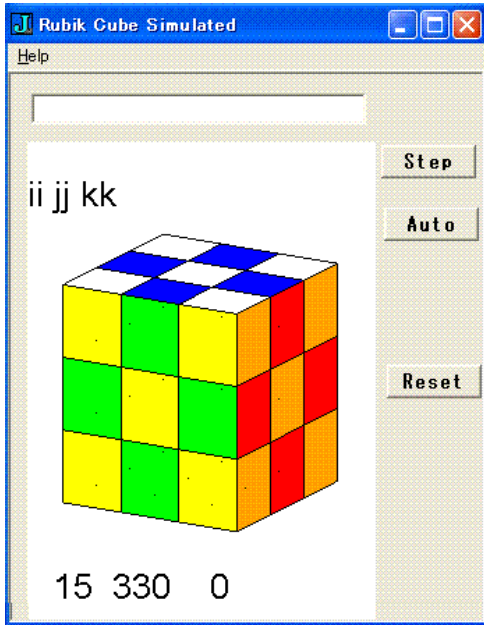
—— 3D立体の把握は難しい！

ひとつずつ直接入力する方法に加えて、あらかじめプログラムとして与えて、自動的に行う機能などを作った。島内先生の本に多数あげられているイディオムを実行するのに必要な機能である。

#### 4. ルービック・キューブのJ-3Dグラフィックスの実際

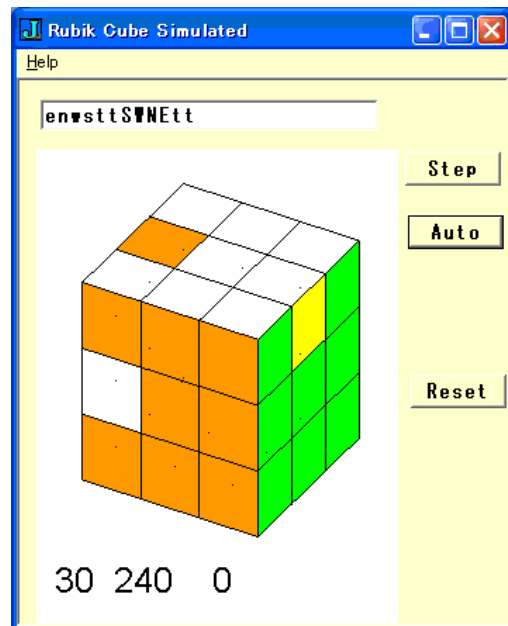
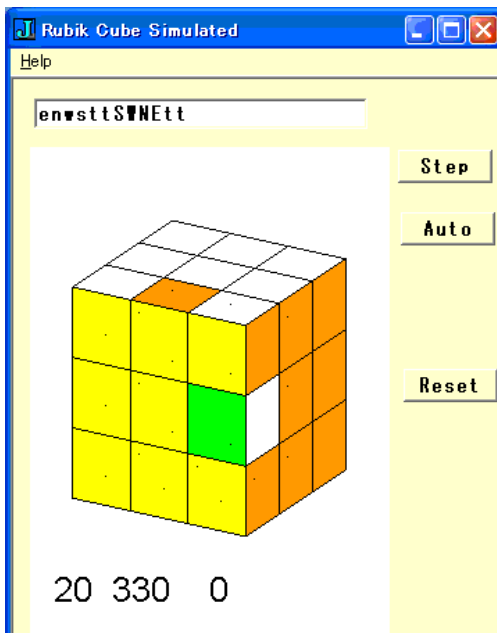
##### 4. 1 簡単なパターン—市松模様と六面

岩倉啓祐「ルービックキューブ」p.112 永岡書店(1981)



##### 4. 2 島内先生の「世界一周」を利用したイディオム「むすび」 [1] p.79

島内先生の本では、いろいろなイディオムが掲載されているが、ルービック操作で影響を受けるキュービーを最低の数とし、さらにすべて明らかにして記載してある。このことから、最も信頼できる攻略本といえる。





## 5. ルービック・キューブ操作の群論へ向けての考察

抽象代数学の一分野である群論の対象には、数、式から図形の対称操作などさまざまなものがある。群論に限ったわけではないが、数学の手法として、数値ではない運動、操作などを式の演算で行うことで、見通しよく効率的に扱うことができる。

島内先生の本では、ルービック・キューブの操作を一種の式で表している。これを参考に式演算で行うシステムをJでプログラムしてみた。

先に定義した色選択の動詞 `sel_color` はキュービーの動きをインデックスの置換とすることで、まさにこれに相当する。これを元に色名の代わりに面の名前を用いて、関数型の動詞 `move1` とし、これを連続して行う動詞 `move` をプログラムした。さらに見やすくするツール `display` を作った。

まず、初期状態を示す。

```
R_DA
+-----+-----+-----+-----+-----+
| WWWWWW | SSSSSSS | EEEEEEE | NNNNNNN | TTTTTTT | BBBBBBB |
+-----+-----+-----+-----+-----+
```

```
display R_DA
+----+
| TTT |
| TTT |
| TTT |
+----+-----+-----+-----+
| WWW | SSS | EEE | NNN |
| WWW | SSS | EEE | NNN |
| WWW | SSS | EEE | NNN |
+----+-----+-----+-----+
| BBB |
| BBB |
| BBB |
+----+
```

先のルービック操作 `'se'` (J と異なり、左から順に) を途中経過を含めて表示する。

```
display (1;R_DA) move 'se'
```

```
0: s
+----+
| TTT |
| TTT |
| WWW |
+----+-----+-----+-----+
| WWB | SSS | TEE | NNN |
| WWB | SSS | TEE | NNN |
| WWB | SSS | TEE | NNN |
+----+-----+-----+-----+
| EEE |
| BBB |
| BBB |
+----+
```

```

-----
1: e
  +---+
  |TTS|
  |TTS|
  |WWS|
+---+---+---+---+
|WWB|SSE|TTT|WNN|
|WWB|SSB|EEE|TNN|
|WWB|SSB|EEE|TNN|
+---+---+---+---+
  |EEN|
  |BBN|
  |BBN|
  +---+
-----

```

```

Result:
  +---+
  |TTS|
  |TTS|
  |WWS|
+---+---+---+---+
|WWB|SSE|TTT|WNN|
|WWB|SSB|EEE|TNN|
|WWB|SSB|EEE|TNN|
+---+---+---+---+
  |EEN|
  |BBN|
  |BBN|
  +---+

```

島内先生のイディオム「直角むすび」をやってみた。(前節 4.2 の図)

```
display R_DA move 'enwsttSWNEtt'
```

```

Result:
  +---+
  |TTT|
  |TTT|
  |TET|
+---+---+---+---+
|WWW|SSS|EEE|NSN|
|WWW|SSN|TEE|NNN|
|WWW|SSS|EEE|NNN|
+---+---+---+---+
  |BBB|
  |BBB|
  |BBB|

```

+---+

## プログラム・リスト

NB. OpGLN\_Rubik.ijs

NB. 2011/9/19

NB. referred from OpGLN3.ijs

```
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 209 23 34 11;cc Step button;
```

```
xywh 210 43 34 11;cc Auto button;
```

```
xywh 7 6 121 11;cc Program edit ws_border es_autohscroll;
```

```
pas 0 0;
```

```
rem form end;
```

```
)
```

NB. Color Definition =====

NB. revised for OpenGL spec. 2011/10/12

```
R =: 1 0 0 NB. red
```

```
O =: 1 0.64 0 NB. 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
```

```
p =: 0 0.8 0.7 NB. sky blue for work
```

```
Q =: 0 0 0 NB. black
```

```
test_C =: 'PGGGGQGG'
```

```
run=: aa_run
```

```
aa_run=: 3 : 0
```

```
wd :: ] 'psel a;pclose'
```

```
wd AA
```

```

glaRC ''
RR =: 0 0 0
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 =====
if. 0 = #y. do.
  S_COL =: 'YYYYYYYYY'
  E_COL =: '000000000'
  W_COL =: 'RRRRRRRRR'
  T_COL =: 'WWWWWWWWW'
  B_COL =: 'BBBBBBBBB'
  N_COL =: 'GGGGGGGGG'
NB. N_COL =: test_C
else.
  init_color y. NB. change color patterns in several books
end.
istep =: 0
wd 'pshow;iptop'
)

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
RR =: 360 | RR + 5 * 'xyz' = 0 { sysdata
RR =: 360 | RR - 5 * 'XYZ' = 0 { sysdata
NB. Change Color of Cubies for Rubik Moves =====
KK0 =. 0 { sysdata
sel_color KK0 NB. move Rubik and change color
KK1 =. KK0 -. 'xyzXYZ'
Key =: Key, KK1 NB. record of moves
glpaintx''
)

```

```

sel_color =: 3 : 0
S_C =. S_COL
T_C =. T_COL
E_C =. E_COL
W_C =. W_COL
N_C =. N_COL
B_C =. B_COL
select. y.
  case. 'e' do.
    E_COL =: , |: |. (3 3)$E_C
    S_COL =: ((2 5 8){B_C} (2 5 8) } S_C
    T_COL =: ((2 5 8){S_C} (2 5 8) } T_C
    N_COL =: ((8 5 2){T_C} (0 3 6) } N_C NB. revised by N-face
    B_COL =: ((0 3 6){N_C} (8 2 5) } B_C NB. revised by N-face
  case. 'E' do.
    E_COL =: , |: |."(1) (3 3)$E_C
    S_COL =: ((2 5 8){T_C} (2 5 8) } S_C
    T_COL =: ((0 3 6){N_C} (8 5 2) } T_C NB. revised by N-face
    N_COL =: ((8 5 2){B_C} (0 3 6) } N_C NB. revised by N-face
    B_COL =: ((2 5 8){S_C} (2 5 8) } B_C
  case. 'w' do.
    W_COL =: , |: |. (3 3)$W_C
    T_COL =: ((8 5 2){N_C} (0 3 6) } T_C NB. revised by N-face
    S_COL =: ((0 3 6){T_C} (0 3 6) } S_C
    B_COL =: ((0 3 6){S_C} (0 3 6) } B_C
    N_COL =: ((0 3 6){B_C} (8 5 2) } N_C NB. revised by N-face
  case. 'W' do.
    W_COL =: , |: |."(1) (3 3)$W_C
    T_COL =: ((0 3 6){S_C} (0 3 6) } T_C
    S_COL =: ((0 3 6){B_C} (0 3 6) } S_C
    B_COL =: ((8 5 2){N_C} (0 3 6) } B_C NB. revised by N-face
    N_COL =: ((0 3 6){T_C} (8 5 2) } N_C
  case. 'n' do.
    N_COL =: , |: |. (3 3)$N_C
    B_COL =: ((0 3 6){W_C} (6 7 8) } B_C
    E_COL =: ((6 7 8){B_C} (8 5 2) } E_C
    T_COL =: ((8 5 2){E_C} (2 1 0) } T_C
    W_COL =: ((2 1 0){T_C} (0 3 6) } W_C
  case. 'N' do.
    N_COL =: , |: |."(1) (3 3)$N_C
    B_COL =: ((8 5 2){E_C} (6 7 8) } B_C
    E_COL =: ((2 1 0){T_C} (8 5 2) } E_C
    T_COL =: ((0 3 6){W_C} (2 1 0) } T_C

```

$W\_COL =: ((6\ 7\ 8)\{B\_C\}\ (0\ 3\ 6))\ } W\_C$   
 case. 't' do.  
 $T\_COL =: ,\ |:\ |.\ (3\ 3)\$T\_C$   
 $N\_COL =: ((2\ 1\ 0)\{W\_C\}\ (2\ 1\ 0))\ } N\_C$  NB. revised by N-face  
 $E\_COL =: ((2\ 1\ 0)\{N\_C\}\ (2\ 1\ 0))\ } E\_C$  NB. revised by N-face  
 $S\_COL =: ((2\ 1\ 0)\{E\_C\}\ (2\ 1\ 0))\ } S\_C$   
 $W\_COL =: ((2\ 1\ 0)\{S\_C\}\ (2\ 1\ 0))\ } W\_C$   
 case. 'T' do.  
 $T\_COL =: ,\ |:\ |.\ "(1)\ (3\ 3)\$T\_C$   
 $N\_COL =: ((2\ 1\ 0)\{E\_C\}\ (2\ 1\ 0))\ } N\_C$  NB. revised by N-face  
 $E\_COL =: ((2\ 1\ 0)\{S\_C\}\ (2\ 1\ 0))\ } E\_C$   
 $S\_COL =: ((2\ 1\ 0)\{W\_C\}\ (2\ 1\ 0))\ } S\_C$   
 $W\_COL =: ((2\ 1\ 0)\{N\_C\}\ (2\ 1\ 0))\ } W\_C$  NB. revised by N-face  
 case. 's' do.  
 $S\_COL =: ,\ |:\ |.\ (3\ 3)\$S\_C$   
 $T\_COL =: ((8\ 5\ 2)\{W\_C\}\ (6\ 7\ 8))\ } T\_C$   
 $E\_COL =: ((6\ 7\ 8)\{T\_C\}\ (0\ 3\ 6))\ } E\_C$   
 $B\_COL =: ((0\ 3\ 6)\{E\_C\}\ (2\ 1\ 0))\ } B\_C$   
 $W\_COL =: ((2\ 1\ 0)\{B\_C\}\ (8\ 5\ 2))\ } W\_C$   
 case. 'S' do.  
 $S\_COL =: ,\ |:\ |.\ "(1)\ (3\ 3)\$S\_C$   
 $T\_COL =: ((0\ 3\ 6)\{E\_C\}\ (6\ 7\ 8))\ } T\_C$   
 $E\_COL =: ((2\ 1\ 0)\{B\_C\}\ (0\ 3\ 6))\ } E\_C$   
 $B\_COL =: ((8\ 5\ 2)\{W\_C\}\ (2\ 1\ 0))\ } B\_C$   
 $W\_COL =: ((6\ 7\ 8)\{T\_C\}\ (8\ 5\ 2))\ } W\_C$   
 case. 'b' do.  
 $B\_COL =: ,\ |:\ |.\ (3\ 3)\$B\_C$   
 $S\_COL =: ((6\ 7\ 8)\{W\_C\}\ (6\ 7\ 8))\ } S\_C$   
 $E\_COL =: ((6\ 7\ 8)\{S\_C\}\ (6\ 7\ 8))\ } E\_C$   
 $N\_COL =: ((6\ 7\ 8)\{E\_C\}\ (6\ 7\ 8))\ } N\_C$  NB. revised by N-face  
 $W\_COL =: ((6\ 7\ 8)\{N\_C\}\ (6\ 7\ 8))\ } W\_C$  NB. revised by N-face  
 case. 'B' do.  
 $B\_COL =: ,\ |:\ |.\ "(1)\ (3\ 3)\$B\_C$   
 $S\_COL =: ((6\ 7\ 8)\{E\_C\}\ (6\ 7\ 8))\ } S\_C$   
 $E\_COL =: ((6\ 7\ 8)\{N\_C\}\ (6\ 7\ 8))\ } E\_C$  NB. revised by N-face  
 $N\_COL =: ((6\ 7\ 8)\{W\_C\}\ (6\ 7\ 8))\ } N\_C$  NB. revised by N-face  
 $W\_COL =: ((6\ 7\ 8)\{S\_C\}\ (6\ 7\ 8))\ } W\_C$   
 NB. move through inner center and edge cubies 2011/10/6  
 case. 'i' do.  
 $S\_COL =: ((3\ 4\ 5)\{W\_C\}\ (3\ 4\ 5))\ } S\_C$   
 $W\_COL =: ((3\ 4\ 5)\{N\_C\}\ (3\ 4\ 5))\ } W\_C$   
 $N\_COL =: ((3\ 4\ 5)\{E\_C\}\ (3\ 4\ 5))\ } N\_C$   
 $E\_COL =: ((3\ 4\ 5)\{S\_C\}\ (3\ 4\ 5))\ } E\_C$   
 case. 'I' do.  
 $S\_COL =: ((3\ 4\ 5)\{E\_C\}\ (3\ 4\ 5))\ } S\_C$

```

W_COL =: ((3 4 5) {S_C} (3 4 5) } W_C
N_COL =: ((3 4 5) {W_C} (3 4 5) } N_C
E_COL =: ((3 4 5) {N_C} (3 4 5) } E_C
case. 'j' do.
S_COL =: ((1 4 7) {B_C} (1 4 7) } S_C
T_COL =: ((1 4 7) {S_C} (1 4 7) } T_C
N_COL =: ((1 4 7) {T_C} (7 4 1) } N_C
B_COL =: ((7 4 1) {N_C} (1 4 7) } B_C
case. 'j' do.
S_COL =: ((1 4 7) {T_C} (1 4 7) } S_C
T_COL =: ((7 4 1) {N_C} (1 4 7) } T_C
N_COL =: ((1 4 7) {B_C} (7 4 1) } N_C
B_COL =: ((1 4 7) {S_C} (1 4 7) } B_C
case. 'k' do.
E_COL =: ((5 4 3) {B_C} (1 4 7) } E_C
T_COL =: ((1 4 7) {E_C} (3 4 5) } T_C
W_COL =: ((3 4 5) {T_C} (7 4 1) } W_C
B_COL =: ((7 4 1) {W_C} (5 4 3) } B_C
case. 'K' do.
E_COL =: ((3 4 5) {T_C} (1 4 7) } E_C
T_COL =: ((7 4 1) {W_C} (3 4 5) } T_C
W_COL =: ((5 4 3) {B_C} (7 4 1) } W_C
B_COL =: ((1 4 7) {E_C} (5 4 3) } B_C
end.
)

```

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 " : 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
  S_COL =: 'YYYYYYYYY'
  E_COL =: '000000000'
  W_COL =: 'RRRRRRRRR'
  T_COL =: 'WWWWWWWWW'
  B_COL =: 'BBBBBBBBB'
  N_COL =: 'GGGGGGGGG'
draw_rubik ''
draw_frames ''
Key =: ''
drawtext ''
istep =: 0
glSwapBuffers ''
wd 'setfocus g'
)

```

```

aa_Program_button=: 3 : 0
PDA =: Program
wd 'setfocus g'
)

```

```

aa_Step_button=: 3 : 0
if. istep < #PDA
  do.
    sel_color istep{PDA
    draw_rubik ''
    draw_frames ''
    glSwapBuffers ''
    istep =: istep + 1
  end.
wd 'setfocus g'
)

```

```

aa_Auto_button=: 3 : 0
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT
  S_COL =: 'YYYYYYYYY'
  E_COL =: '000000000'
  W_COL =: 'RRRRRRRRR'
  T_COL =: 'WWWWWWWWW'
  B_COL =: 'BBBBBBBBB'
  N_COL =: 'GGGGGGGGG'
sel_color L:0 <"(0) PDA
draw_rubik ''
draw_frames ''

```

```

Key =: ''
drawtext ''
PDA =: ''
glaSwapBuffers ''
wd 'setfocus g'
)
NB. Calc. Cubie Points =====
Point =: |. |: {(i:1);(i:1)

NB. Along -Z axis (looking far), square_numbered counter_clockwise! ====
boundary =: 3 : 0
'X Y' =. y.
((0.5+X), 0.5+Y);((-0.5+X), 0.5+Y);((-0.5+X), _0.5+Y);((0.5+X), _0.5+Y)
)

Bound =. >, boundary L:0 Point
S_P =: (&1.5) L:0 Bound

E_P =: |. L:0 Bound
E_P =: ({. , (-@{:)) L:0 E_P
E_P =: (1.5&,) L:0 E_P

W_P =: |. L:0 Bound
W_P =: (_1.5&,) L:0 W_P

T_P =: ({. , (-@{:)) L:0 Bound
T_P =: (1.5&,) L:0 T_P
T_P =: ((1 0 2)&{}) L:0 T_P

B_P =: (_1.5&,) L:0 Bound
B_P =: ((1 0 2)&{}) L:0 B_P

N_P =: ((-@{.), {:) L:0 Bound
N_P =: (&_1.5) L:0 N_P

NB. Set Color of Cubies =====

face_col =: 3 : 0
:
i =. 0
while. i < 9 do.
  ("i{x.) polygon >i{y.
  i =. i + 1
end.
)

```

```

polygon=: 4 : 0
glColor x.
glBegin GL_POLYGON
  glVertex y.
glEnd ''
)
draw_rubik =: 3 : 0
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glTranslate 0 0 _1
glRotate RR ,. 3 3 $ 1 0 0 0
glPolygonMode GL_FRONT, GL_FILL NB. Front and Back: Full Paint
glPolygonMode GL_BACK, GL_POINT NB. Back: Point (Hidden)
glBegin GL_QUADS
(>B_COL) face_col (0.6&*) L:0 B_P NB. reduced 0.6 size
(>W_COL) face_col (0.6&*) L:0 W_P
(>T_COL) face_col (0.6&*) L:0 T_P
(>E_COL) face_col (0.6&*) L:0 E_P
(>N_COL) face_col (0.6&*) L:0 N_P
(>S_COL) face_col (0.6&*) L:0 S_P
glEnd ''
)
NB. Frame Cubies == 2011/9/23 =====
face_frame =: 3 : 0
i =. 0
while. i < 9 do.
  fpolygon i{y.
  i =. i + 1
end.
)

fpolygon=: 3 : 0
glColor 0 0 0
glBegin GL_POLYGON
  glVertex y.
glEnd ''
)

draw_frames =: 3 : 0
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glTranslate 0 0 _1
glRotate RR ,. 3 3 $ 1 0 0 0
glPolygonMode GL_FRONT, GL_LINE NB. Front and Back: Full Paint

```

```
glPolygonMode GL_BACK, GL_POINT NB. Back: Point (Hidden)
face_frame > (0.6&*) L:0 S_P
face_frame > (0.6&*) L:0 T_P
face_frame > (0.6&*) L:0 E_P
face_frame > (0.6&*) L:0 W_P
face_frame > (0.6&*) L:0 N_P
face_frame > (0.6&*) L:0 B_P
)
```

NB. Rubik\_S.ijs 2011/10/19 by T. Nishikawa  
 NB. referred from OpGLN\_Rubik.ijs  
 NB. expresion based on "Rubik" by Shimanouchi

NB. Usage:

NB. display R\_DA => 初期状態表示  
 NB. display R\_DA move 's' => ルービック操作's'の結果表示  
 NB. display R\_DA move 'enws' => 最終結果のみ表示  
 NB. display (0;R\_DA) move 'enws' => 最終結果のみ表示  
 NB. display (1;R\_DA) move 'enws' => 途中経過表示

R\_DA =: (9&#) L:0 'W';'S';'E';'N';'T';'B'

```
display =: 3 : 0
RD0 =. (3, 3)$ L:0 (4{. y.)
RD1 =. (3, 3)$ L:0 (9#' ');(>4{. 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
)
```

wr =: 1!:2&2

```
move =: 3 : 0
:
if. 6 = #x. do. dis =: 0 [ D =. x. end.
if. 7 = #x. do. dis =: > {. x. [ D =. }. x. end.
i =. 0
while. i < #y.
do.
  D =. D move1 i{y.
  if. 1 = dis do.
    wr (":i), ': ', (":i{y.)
    wr display D
    wr '-----'
  end.
  i =. i + 1
end.
wr 'Result:'
D
```

```

)
move1 =: 3 : 0
NB. modified from sel_color
:
'W_C S_C E_C N_C T_C B_C' =. x.
'W_COL S_COL E_COL N_COL T_COL B_COL' =. x.

select. y.
  case. 'e' do.
    E_COL =. , |: |. (3 3)$E_C
    S_COL =. ((2 5 8){B_C} (2 5 8) } S_C
    T_COL =. ((2 5 8){S_C} (2 5 8) } T_C
    N_COL =. ((8 5 2){T_C} (0 3 6) } N_C NB. revised by N-face
    B_COL =. ((0 3 6){N_C} (8 2 5) } B_C NB. revised by N-face
  case. 'E' do.
    E_COL =. , |: |."(1) (3 3)$E_C
    S_COL =. ((2 5 8){T_C} (2 5 8) } S_C
    T_COL =. ((0 3 6){N_C} (8 5 2) } T_C NB. revised by N-face
    N_COL =. ((8 5 2){B_C} (0 3 6) } N_C NB. revised by N-face
    B_COL =. ((2 5 8){S_C} (2 5 8) } B_C
  ---
  (途中省略)
  ---
end.
W_COL;S_COL;E_COL;N_COL;T_COL;B_COL
)

```