J-OpenGL によるルービック・キューブの３Ｄグラフィックス－４
－色コマンド操作および定石と攻略の実際－
付　プログラムの構成と J6 への変更

西川　利男

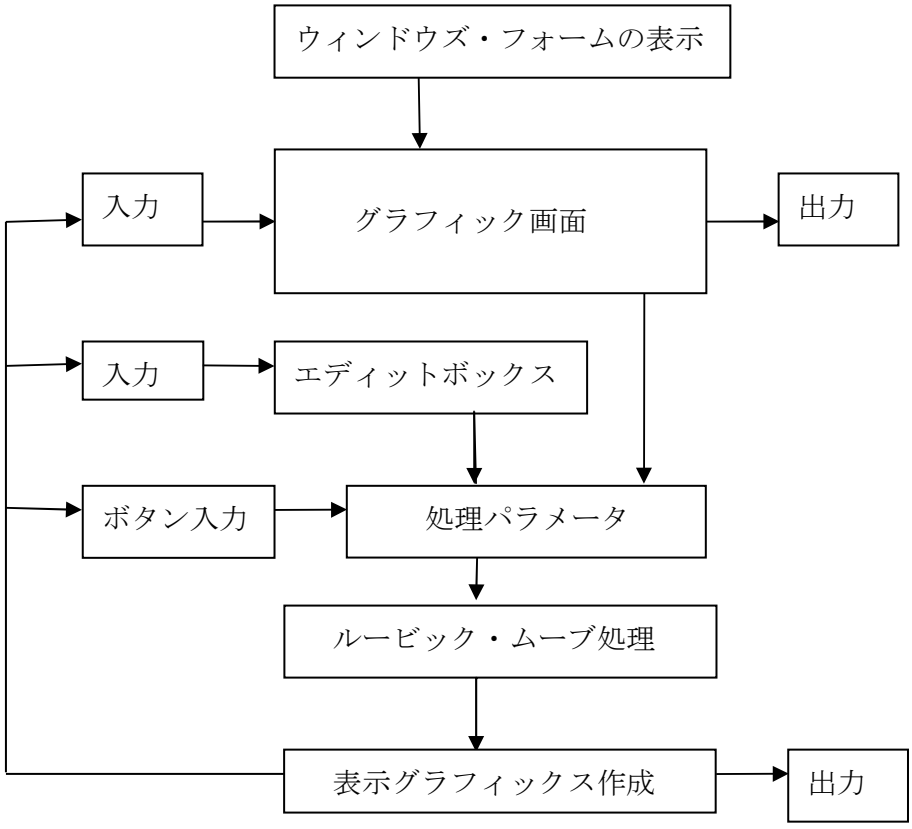
これまでは、ルービック・キューブの処理プログラムを作るのに精一杯であったが、ようやく実用のシステムになりつつある。このようなシステムでは、ユーザ入力環境が極めて大切である。つまり、値を入れて計算結果を得ればよい、というものではなく、ユーザ入力に対して画面に表示し、その結果を見て次の入力を行い、…というやり取りを続ける、というシステムである。このようなユーザ・インターフェースを整備して、より使いやすい実用システムの構築を目指している。

1.　システムの特徴と構成

システムのポイントは次の３つに要約されよう。

・ルービック・キューブの動きを群論の置換操作で擬似実験
・J のプリミティブ C. を利用した効果的プログラミング
・OpenGL の立体グラフィック・アニメーションによる視覚化

システムの構成は次のようである。



２．ルービック面とルービック操作の方式

このシステムではルービック･キューブの小片(cubelet とも呼ばれる)の位置を名前で示し、操作コマンドにより、その色を替えることで、ルービック･キューブの動きを実現する。ユーザ・フレンドリという観点から、ユーザの入力は極めて大切である。ここでルービック･キューブのデータ構造の命名などを、改めて見直してみよう。

Ｊの見方からはルービック面の名前は名詞であり、ルービック操作は動詞である。使い易さという点からは、この２つは対応させた方がよい。

ルービック･キューブを立体図形と見たとき、座標の取り方には２通りある。

・固定座標（絶対座標、世界座標）…… アルゴリズムの記述に便

・視点座標（相対座標、見かけのローカル座標）… ユーザの実際の操作に便

これまでの方式を見てみると

W, S, E, N, T, B （島内方式）…… 絶対座標

L, F, R, B, U, D （Joyner 方式）… 本来は視点座標であるのが、絶対座標として使われている。

色指定方式 …… 絶対座標でありながら、ユーザの視点をもとに扱えるので、最もユーザ・フレンドリである。したがって、手軽に使うには色指定のコマンドが良い。

しかしながら、プログラミングにあたっては、以下の問題点もある。

・コマンドはグローバルの名前となるため、既存の名前との衝突、競合が起きる。

・プログラム・グローバルとウィンドウ・グローバルとで切り抜けられないか。

結局、ウィンドウ画面でのキー・コマンドの名前として、以下のようにした。

色指定コマンド… B(Blue), K(Ki-iro), R(Red), G(Green), O(Orange), W(White)

中抜き回転…… S(Side), M(Middle), E(Equatorial)

ルービック画像の回転… X, x, Y, y, Z, z

アルゴリズム記述とルービック操作 … いままで通りの Joyner 方式

ごらんのように競合のため、色指定コマンドとして、Y(Yellow)とは出来なかった。

実際のプログラムでは、ウィンドウ画面での入力キー・コマンドとルービック操作との間で、名前の変更、復帰を行っている。

３．ルービック・キューブの攻略法の流れと定石

ルービック・キューブの攻略法にはいろいろあるが、ここでは一番標準的なコーナー・エッジ法をとりあげた。これは「宝島社」の攻略本ではツクダ式として操作が丁寧に図で説明されている。

その攻略法の大きな流れは、上段をはじめに揃えて、次に下段を揃え、最後に中段を揃えて完成させる。

(1.1) 上段のコーナ・キューブの位置と向きを揃える。必要に応じて下段から持ってくる。

(1.2) 上段のエッジ・キューブの位置と向きを揃える。必要に応じて下段、中段から持ってくる。

(2.1) 下段のコーナ・キューブの位置を揃える。ついで向きを揃える。

(2.2) 下段のエッジ・キューブの位置と向きを揃える。

(3) 中段のエッジ・キューブの位置と向きを揃える。

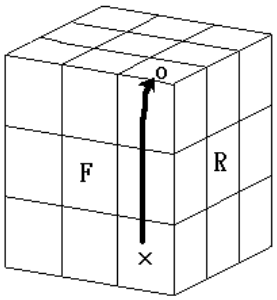
最初は上段のキューブを揃えるとき、中段、下段のキューブの位置、向きに相当影響がでるが(副作用)、これは無視してよい。しかし、下段、中段では、いままでそろ

ったキューブに影響を与えないよう移動することが必要になる。そのようなことも含めて、多くのキューブ移動の定石がある。

４． キューブ移動のアルゴリズムとその考え方

４．１　正面(F)右下(dfr) のコーナ・キューブを上面(U)右下(ufr)へ移動

キューブレットの表し方は、Joyners の本にある Singmaster 記法にしたがった。



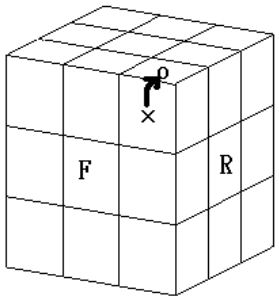
このとき、上面の他のキューブは変えず、元の状態が保たれる。中段、下段のキューブには影響する。

これは、次の操作で行うことができる。

Ｒｆｒ Ｆ

R　Ｒ面で時計周りの回転 ……目的位置へのキューブを移動
f　Ｆ面で反時計周りの回転 … 移動したキューブを退避
r　Ｒ面で反時計周りの回転 … 元の状態への復元
F　Ｆ面で時計周りの回転 …… 退避したキューブの復元

4.2　正面(F)右上(ufr) のコーナ・キューブを上面(U)右下(ufr)へ移動



この例では、位置を変えず、向きだけ変える。つまりひねりをやっている。

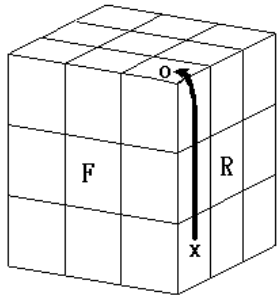
これは、次の操作で行うことができる。

Ｒｆｒ Ｆ　Ｒｆｒ Ｆ

最初の操作Ｒｆｒ Ｆでは位置(ufr)のコーナ・キューブを位置(dfr)に移す。すると、４．１の状態になるので、さらにＲｆｒ Ｆの操作を行う。

４．３　右面(R)左下(dfr)のコーナ・キューブを上面(U)右下(ufr)へ移動

この操作の結果は、４．１の場合と異なり、コーナ・キューブの向きが違う。



これは、次の操作で行うことができる。

ｆＲＦｒ

f　Ｆ面で反時計周りの回転 …目的位置へのキューブの移動
R　Ｒ面で時計周りの回転 ……移動したキューブを退避
F　Ｆ面で時計周りの回転 ……元の状態への復元
r　Ｒ面で反時計周りの回転 …退避したキューブの復元

これらは攻略の最初の段階で必要な、最も基本となるキューブ移動の操作である。、キューブの移動はこのような考え方で、さまざまな一種の定石が使われる。

なお、これらの操作を行うと、その結果として、他のキューブレットにもいろいろ影響を与えるので、これに留意しておくことが必要である。

以上の３つの操作は、それぞれはつぎのようなプログラムとした。

```
 1 cmov 'FR'
 2 cmov 'FR'
_1 cmov 'FR'
```

以下、いろいろなキューブ移動の定石とプログラムを示す。

５．いろいろなキューブ移動の定石とプログラム

５．１　上段へのコーナ・キューブの移動［1］p.13

```
 1 cmov 'FR'  ←→ R f r F                    正面(F)dfrから上面(U)ufrへ
 2 cmov 'FR'  ←→ R f r F  R f r F           ufrのcwひねり
 3 cmov 'FR'  ←→ R f r F  R f r F  R f r F  下面(D)dfrから上面(U)ufrへ
_1 cmov 'FR'  ←→ f R F r                    右面(R)dfrから上面(U)ufrへ
_2 cmov 'FR'  ←→ f R F r  f R F r           ufrのccwひねり
```

中段、下段に影響を与える。

５．２　上段へのエッジ・キューブの移動［1］p.14, 15

```
 1 emov 'R'  ←→ R s r S                     正面(F)frから右上面(U)urへ
 2 emov 'R'  ←→ R R s r r S                 下面(D)drから右上面(U)urへ
 3 emov 'R'  ←→ R R R s r r r S             裏面(B)brから右上面(U)urへ
 4 emov 'R'  ←→ s R S r                     右下面(R)drから右上面(U)urへ
 5 emov 'R'  ←→ R E r r E E R E             右上面ur自身の向きの交換
```

中段、下段に影響を与える。

５．３　下段、各面の両脇のコーナ・キューブの位置の交換［1］p.17

```
cpos 'F'   ←→ b D B r B R b D D             前面(fldとfrd)の位置交換
cpos 'R'   ←→ l D L b L B l D D             右面(rfdとrbd)の位置交換
cpos 'B'   ←→ f D F l F L f D D             裏面(brdとbld)の位置交換
cpos 'L'   ←→ r D R f R F r D D             左面(lbdとlfd)の位置交換
```

上段へは影響なし、中段に影響を与える。

５．４　下段、１つを不変として３つのコーナ・キューブの向きの交換［1］p.19

```
cdir 'F'  ←→ f D D F D f D F D D            frdは不変、それ以外をねじる
cdir 'R'  ←→ r D D R D r D R D D            rbdは不変、それ以外をねじる
cdir 'B'  ←→ b D D B D b D B D D            bldは不変、それ以外をねじる
cdir 'L'  ←→ l D D L D l D L D D            lfdは不変、それ以外をねじる
```

上段、中段は変わらず、下段の３つのコーナ・キューブ向きだけが変わる。

５．５　下段のエッジ部分に中段からのエッジ・キューブを移動［1］p.21

```
 1 dfrch 'R' ←→ R e r E r e R   R面をコーナとする右エッジ・キューブを移動
_1 dfrch 'R' ←→ r E R e R E r   R面をコーナとする左エッジ・キューブを移動
 2 dfrch 'R' ←→ R e r r e R e   r E R e R E r  下段エッジ・キューブの交換
```

上段は変わらず、中段、下段が変わる。

５．６　中段のエッジ・キューブの位置の巡回置換［1］p.23

```
epos 'F'  ←→ F F e F F E       lbだけ不変、それ以外のエッジを巡回置換
epos 'R'  ←→ R R e R R E       flだけ不変、それ以外のエッジを巡回置換
epos 'B'  ←→ B B e B B E       frだけ不変、それ以外のエッジを巡回置換
epos 'L'  ←→ L L e L L E       rbだけ不変、それ以外のエッジを巡回置換
```

上の操作では、他のキューブレットには影響を与えない。

５．７　中段の２つのエッジ・キューブの向きの交換［1］p.25
＝ルービック・マヌーバ

```
rubm 'RE' ←→ E R E R E R R   e R e R E R R
rubm 'FE' ←→ E F E F E F F   e F e F E F F
```

上の操作では、他のキューブレットには影響を与えない。

６．ルービック・キューブ操作と解への攻略の実際

６．１　いろいろな起動の方法とテストのオプション

・通常の起動

　run ’’

・テストのための起動…引数 initC により指定

　run initS(＝initC, ColInp)

initC(第１引数) … 色の設定

　initC: 10 … 「宝島社」の攻略本、Joyner 方式の色設定

　initC: 0 … 島内の色設定

　initC: 21 … ルービック・マヌーバのテスト用の色設定

　initC: 91 … 色マーク(水色)でキューブレットの位置指定(入力ウィンドウで)

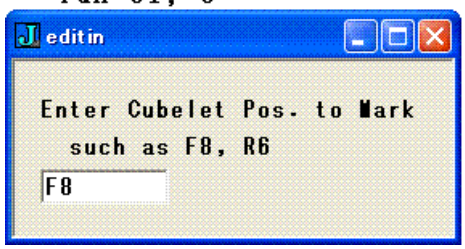
ColInp(第２引数) … ウィンドウ画面上でのコマンド入力の方法

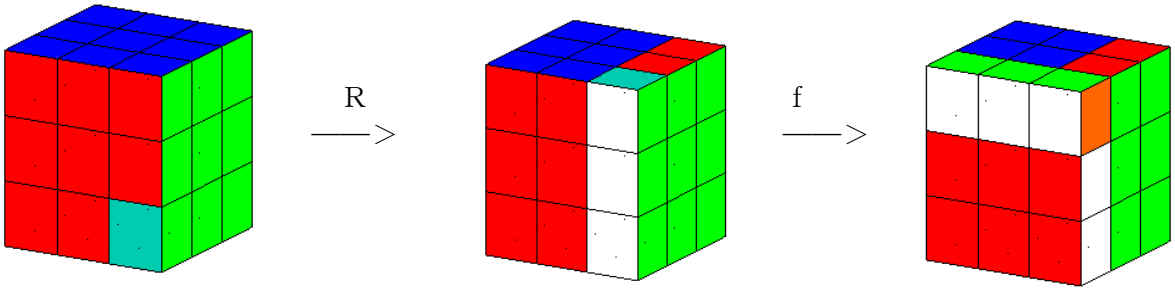
　ColInp: 0 … 「宝島社」の攻略本、Joyner 方式

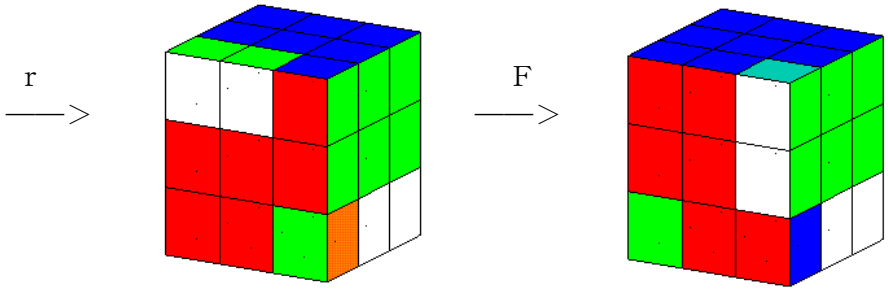
　ColInp: 1 … 色指定方式（３．を参照）

例として、色マーク(水色)オプションを用いて、先の４．１のルービック操作の途中経過を示してみよう。

```
run 91, 0
```





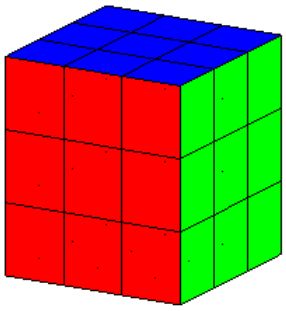


６．２ ルービック・キューブ攻略の実際

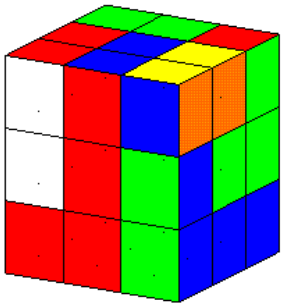
ここでは

run 10, 1

として、色指定で行ってみる。また、乱数により右図のような課題を作ってみた。

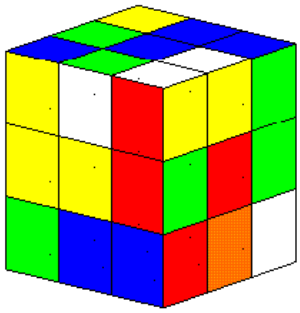


rnd 8(gkRWOrGW)
---------->

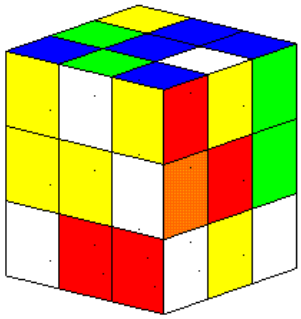


いろいろ動かし、上段、青面が、左図のようになった。次に黄色面の右下コーナ・キューブ青を上段に移動したい。ここで、５．１の定石を使って移動させると、右図のようになる。

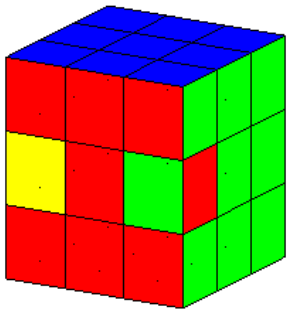
1



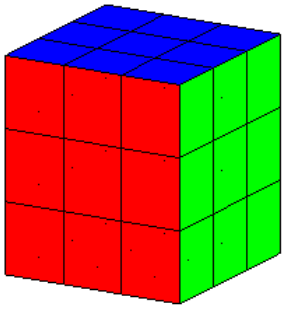
cmov 'KR'
---------->



さらに、３．の攻略法にしたがって、下段のコーナ、エッジ、最後に中段と揃えて行く。大切なことは、この段階ではすでに揃ったキューブレットの配置を崩さないで、進めて行かなくてはならない。そのためにも５．の定石が必要である。ルービック・マヌーバと呼ばれるすばらしい定石の効果を示す。



rubm 'RE'
ERERERR eReRERR
---------->



付１．OpGLN_RubikJ3.ijs のプログラム構成

・キューブレットの色指定
　各種の起動に対応
・AA　ウィンドウフォーム設定値
・run　=　aa_run　ウィンドウフォームの表示とプログラム起動

OpenGL による起動時自動実行
aa_paint … OpenGL 画面の表示
draw_rubik
face_col 色データ　poygon　3D 空間上で正方形描画
draw_frame
face_frame　　fpolygon　3D 空間上で正方形輪郭

aa_char … OpenGL 画面上でのキー入力と関連操作
表示画像の回転 … x, X, y, Y, z, Z
ルービック操作(Joyner) … u, U, l, L, f, F, r, R, b, B, d, D;
中抜き回転 … m, M, s, S, e, E
ルービック操作(色指定) … b, B, k, K, r, R, g, G, o, O, w, W
col2fac(色入力用), colx(色出力用)で変換
sel_col … ルービック操作プログラム
「ルービック・キューブの色の動き＝群の置換操作」
色表示文字列 RUBX を J のプリミティブ C. で置換演算を行う

aa_size … OpenGL 画面の大きさの設定
・エディットボックス入力　…
プログラムの入力 Program
・ボタン実行　…
COMPILE … プログラムのコンパイル(calc)
AUTO　…… 一括実行
STEP　…… ステップ実行
CLEAR

付２．J4 から J6 プログラム(OpGLN_RubikJ6.ijs)への変更のポイント

１．J4→J6 の仕様の差によるもの

動詞の引数　y.→y, x.→x, u.→u, v.→v
名詞の定義　グローバル(=:)に

２．J4(直接の OpenGL)から J6(オブジェクト指向の OpenGL)へ

インクルード・ファイル

```
(J4)  require 'gl3' ──→ (J6) require 'opengl gl3'
                              coinsert 'jgl3'
```

フォーム

```
(J4)  cc g isigraph ws_clipchildren ws_clipsiblings
(J6)  cc g isigraph opengl
```

実行プログラム

```
run
  (J4)  glaRC ''
  (J6)  ogl =: '' conew 'jzopengl'

  (J4)                          (J6)
aa_paint =:                  g_paint =:
   draw_rubik ''             g_draw_rubik ''
   draw_frames ''      ──→   g_draw_frames ''
   drawtext ''               g_drawtext ''
   glaSwapBuffers ''         show__ogl ''

   glColor             ──→   glColor4D
```

プログラム・リスト

```
NB. OpGLN_RubikJ3.ijs  2012/3/7
NB.   Rename T, W, S, E, B, N -> U, L, F, R, D, B
NB.   run 31, 0 => test F8
NB.   setrub 'F8', run 91, 0 => Facet をマーク(水色)での実行
NB.   from OpGLN_RubikJ2.ijs
NB.   from OpGLN_RubikJ.ijs
NB. Rubik Cube Simulation in Joyner's Group Theory Notation (use cycle)
NB. revised 2012/1/17
NB. program <calc '(ER)3R(eR)3R'> OK! 2012/1/20
NB. NJoyF-function checked OK! & revised 2012/1/23

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. run 10, 0 => 宝島社「頭を鍛える」色設定 + 操作コマンドJoyner方式
NB. run 10, 1 => 宝島社「頭を鍛える」色設定 + 操作コマンド色表示
NB. run 21, 0 =>  ''   ルービック・マヌーバのテスト p.25
NB. run 31, 0 => test F8

NB. Facet をマーク(水色)での実行
NB. setrub 'F8', setrub 'R6' => facetの位置を指定
NB. run 91, 0 => mark facet で実行



NB. Attention! Global Nouns of Colors:
NB.  R(red), O(orange), Y(yellow), G(green), W(white), B(blue)

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
```

```
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. 'u' do. 'b'
  case. 'L' do. 'Y'
  case. 'l' do. 'y'
  case. 'F' do. 'R'
  case. 'f' do. 'r'
  case. 'R' do. 'G'
  case. 'r' do. 'g'
  case. 'B' do. 'O'
  case. 'b' do. 'o'
  case. 'D' do. 'W'
  case. 'd' do. 'w'
  fcase.    do.  y.
end.
)

NB. Idiom Compile 色出力用 2012/2/8
colx =: 3 : 0"0
select. y.
  case. 'U' do. 'B'
  case. 'u' do. 'b'
  case. 'L' do. 'K'
```

```
  case. 'l' do. 'k'
  case. 'F' do. 'R'
  case. 'f' do. 'r'
  case. 'R' do. 'G'
  case. 'r' do. 'g'
  case. 'B' do. 'O'
  case. 'b' do. 'o'
  case. 'D' do. 'W'
  case. 'd' do. 'w'
  fcase.    do.  y.
end.
)
NB. Facets from Colors   コマンドの色入力用 2012/2/7
col2fac =: 3 : 0"0
select. y.
  case. 'B' do. 'U'  NB. B(Blue)   => 'U'
  case. 'b' do. 'u'
  case. 'K' do. 'L'  NB. K(Ki-iro, Yellow) => 'L'
  case. 'k' do. 'l'
  case. 'R' do. 'F'  NB. R(Red)    => 'F'
  case. 'r' do. 'f'
  case. 'G' do. 'R'  NB. G(Green)  => 'R'
  case. 'g' do. 'r'
  case. 'O' do. 'B'  NB. O(Orange) => 'B'
  case. 'o' do. 'b'
  case. 'W' do. 'D'  NB. W(White)  => 'D'
  case. 'w' do. 'd'
  fcase.    do. y.
end.
)


NB. Color Definition / Global Nouns ! ===============
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

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

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

NB. Facet Definition ================================
ALL_COJ =: 9#'BYRGOW'  NB. Actual Rubik'S Color Set
RUBJ    =: 9#'ULFRBD'  NB. for Joyner's Facet Name
MRUBJ =: (6, 9)$RUBJ

NB. test_C =: 'PGGGGGQGG'
NB. test for Rubik Manueaver
NB. display T_RUBMAN
T_RUBMAN =:           'UUUUUUUUU'  NB. Up
T_RUBMAN =: T_RUBMAN, 'LLLLLLLLL'  NB. Left
T_RUBMAN =: T_RUBMAN, 'FFFLFRFFF'  NB. Front
T_RUBMAN =: T_RUBMAN, 'RRRFRRRRR'  NB. Right
T_RUBMAN =: T_RUBMAN, 'BBBBBBBBB'  NB. Back
T_RUBMAN =: T_RUBMAN, 'DDDDDDDDD'  NB. Down

NB. display T_F8
T_F8 =:       'UUUUUUUUU'
T_F8 =: T_F8, 'LLLLLLLLL'
T_F8 =: T_F8, 'FFFFFFFFP'
T_F8 =: T_F8, 'RRRRRRRRR'
T_F8 =: T_F8, 'BBBBBBBBB'
T_F8 =: T_F8, 'DDDDDDDDD'

NB. enter Marked Cubelet on edit box 2012/3/9 ======================
edit_in =: 3 : 0
wd 'pc editin;'
wd 'xywh 8 10 120 8;cc s0 static;cn "Enter Cubelet Pos. to Mark";'
wd 'xywh 8 20 120 8;cc s0 static;cn "  such as F8, R6";'
wd 'xywh 8 30 40 10;cc e0 edit;'
wd 'pas 8 8;pcenter;pshow;wait;'
wd 'pclose;'
EditDA =: > {: 12 { wd 'q;'
setrub EditDA
empty ''
)

NB. e.g. setrub 'F8', setrub 'R6', etc.
setrub =: 3 : 0
'C J' =. y.
i =. 'ULFRBD' i. C
j =. ". J
MRUBX =: 'P' (<i;j) } MRUBJ    NB. Global
RUBA =: , MRUBX                NB. Global
```

```
)


NB. ========================================================

run=: aa_run
aa_run=: 3 : 0
initS =: y.
if. 0 = #initS do. goto_skip. end.
if. 91 = {. initS do. edit_in '' end.  NB. Input Marked Cubelet
label_skip.
wd :: ] 'psel a;pclose'
wd AA
glaRC ''
ROT =: 0 0 0  NB. rename from RR
Key =: ''
NL =: 0       NB. NL(New Line) Flag = 0
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 ''  NB. include Joyner's Rubix commands
NB. select command facet or color 2012/2/7
if. 0 = #initS do.
 initS =: 10, 0 NB. デフォルト＝宝島社「頭を鍛える」色設定
end.
initC =: {. initS
NB. for TEST RUBMAN 2012/3/1
if. initC = 21 do. RUBX =: ,> T_RUBMAN end.
if. initC = 31 do. RUBX =: ,> T_F8     end.
if. initC = 91 do. RUBX =: RUBA        end.
ColInp =: {: initS NB. Select Color Input 0: Facet, 1: Color

init_color initC  NB. change color patterns in several books
istep =: 0
PDA =: ''
RecDA =: ''

wd 'pshow;ptop'
)


init_color =: 3 : 0
  select. y.
```

```
    case. 0 do. NB. 島内　色設定
      U_COL =: 'WWWWWWWWW'
      L_COL =: 'RRRRRRRRR'
      F_COL =: 'YYYYYYYYY'
      R_COL =: 'OOOOOOOOO'
      B_COL =: 'GGGGGGGGG'
      D_COL =: 'BBBBBBBBB'
    case. 1  do. NB. 宝島社「頭を鍛える」色設定
      U_COL =: 'BBBBBBBBB'
      L_COL =: 'YYYYYYYYY'
      F_COL =: 'RRRRRRRRR'
      R_COL =: 'GGGGGGGGG'
      B_COL =: 'OOOOOOOOO'
      D_COL =: 'WWWWWWWWW'
    case. 10 do. NB. adjust to Joyners notation
      U_COL =: col 9#'U'
      L_COL =: col 9#'L'
      F_COL =: col 9#'F'
      R_COL =: col 9#'R'
      B_COL =: col 9#'B'
      D_COL =: col 9#'D'
   case. 21 do. NB. to Rubik Maneuver
      U_COL =: 'BBBBBBBBB'
      L_COL =: 'YYYYYRYYY'
      F_COL =: 'RRRYRGRRR'
      R_COL =: 'GGGRGGGGG'
      B_COL =: 'OOOOOOOOO'
      D_COL =: 'WWWWWWWWW'
   case. 31 do. NB. to test F8
      U_COL =: 'BBBBBBBBB'
      L_COL =: 'YYYYYYYYY'
      F_COL =: 'RRRRRRRRP'
      R_COL =: 'GGGGGGGGG'
      B_COL =: 'OOOOOOOOO'
      D_COL =: 'WWWWWWWWW'
   case. 91 do.
      U_COL =: col 0 { MRUBX
      L_COL =: col 1 { MRUBX
      F_COL =: col 2 { MRUBX
      R_COL =: col 3 { MRUBX
      B_COL =: col 4 { MRUBX
      D_COL =: col 5 { MRUBX
  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 ''
NB. drawface1 texdraw pat_d1  NB. 2012/3/6
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
sel_col KK0 NB. move Rubik and change color / new
KK1 =: KK0 -. 'xyzXYZ'
NB. sel_color KK0   NB. move Rubik and change color / old
if. '-' e. Key do. NL =: 1 end.
Key =: Key, KK1 NB. record of moves
RecDA =: RecDA, KK1  NB. RecDA is all record (Key + Program)
glpaintx''
)

NB. new version 2012/1/17 ===================================
sel_col =: 3 : 0
 if. '-' e. y. do. NL =: 1 return. end.  NB. NL(New Line) Flag = 1
 if. +/ 'xyzXYZ' = y. do. return. end.
 KK0 =: y.
 if. 1 = ColInp do.  KK0 =: col2fac KK0 end. NB. select Color Input
 KK =: 2#KK0 NB. KK is Rubik command such as ff, FF, mm, MM, ...
 RUBX =: ". KK,  ' RUBX'  NB. move Rubik facets
 NB. Coloring Facets
 'U_COL L_COL F_COL R_COL B_COL D_COL' =: <"(1) 6 9$ col RUBX
)
```

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


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 =. 0 18 45 44     NB. revised 2012/1/30
l4 =. 3 21 48 41     NB. revised 2012/1/30
l5 =. 6 24 51 38     NB. revised 2012/1/30
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
NB. if NL = 1 with Key = '-', write two lines 2012/1/31
if. NL = 0
  do.
     glRasterPos _2.3 2.1 0
     glCallLists Key
  else.
     lin2mul Key  NB. revised multilines 2012/2/1
end.
glRasterPos _2 _2.4 0
glCallLists (5 ": ROT)  NB. rename from RR
)

NB. one line to multilines punctuated by '-' 2012/2/1
lin2mul =: 3 : 0
LIN =. y.
jj =. 0
while. '-' e. LIN
  do.
    ii =. >: LIN i. '-'
    LIA =. ii{. LIN
    glRasterPos _2.3, (2.1-0.3*jj), 0
    glCallLists LIA
    LIB =. ii}. LIN
    LIN =. LIB
    jj =. jj + 1
end.
glRasterPos _2.3, (2.1-0.3*jj), 0
```

```
glCallLists LIB
)

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
NB. init_color initC
RUBX =: ,> RUBJ
if. initC = 21 do. RUBX =: ,> T_RUBMAN end.
if. initC = 31 do. RUBX =: ,> T_F8     end.
if. initC = 91 do. RUBX =: RUBA        end.

'U_COL L_COL F_COL R_COL B_COL D_COL' =: <"(1) 6 9$ col RUBX

ColInp =: {: initS NB. Select Color Input 0: Facet, 1: Color

draw_rubik ''
draw_frames ''
Key =: ''
drawtext ''
istep =: 0
glaSwapBuffers ''
wd 'set Program ""'
iPDA =: ''
wd 'setfocus g'
)

aa_Program_button=: 3 : 0
PDA =: Program
iPDA =: ''
RecDA =: RecDA, ('.PDA')  NB. All Record (PDA + Key)
wd 'setfocus g'
)

NB. Compile 2012/1/8 ========================
NB. e.g. On Edit Box, Enter "rubm 'RE' CR",
NB.      afterward push Compile button,
NB.      so Rubik Move Program, as follows, will appear on Edit Box
NB.         E R E R E R R e R e R e R R
```

```
NB.      this Program can run by Auto, Step Buttons.
aa_Compile_button=: 3 : 0
wd 'set Program *', '. PDA
wd 'setfocus g'
)


iPDA =: ''
aa_Step_button=: 3 : 0
if. istep < #PDA
  do.
    sel_col istep{PDA
    iPDA =: iPDA, istep{PDA
    wd 'set Program *', iPDA
    wd 'setfocus g'
    draw_rubik ''
    draw_frames ''
    glaSwapBuffers ''
    istep =: istep + 1
  else.
    wd 'set Program "-program end-"'
  end.
wd 'setfocus g'
)

aa_Auto_button=: 3 : 0
wd 'set Program *', PDA
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT
init_color initC
sel_col L:0 <"(0) PDA
draw_rubik ''
draw_frames ''
Key =: ''
drawtext ''
glaSwapBuffers ''
wd 'setfocus g'
)

aa_ClearProg_button=: 3 : 0
PDA =: ''
wd 'set Program " "'
wd 'setfocus g'
)
```

```
NB. Calc. Cubie Points ============================================
Point =: |. |: {(i:1);(i:1)

NB. Along -Z axis (looking far), square_numbered counter_clockwise! ====
boundary0 =: 3 : 0  NB. old version
'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)
)
NB. Bound =:  >, boundary0 L:0 Point

boundary =: 3 : 0    NB. revised for texture 2012/3/6
'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  NB. for Texture 2012/3/6

F_P =: (,&1.5) L:0 Bound

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

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

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

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

B_P =. ((-@{.), {:) L:0 Bound
B_P =: (,&_1.5) L:0 B_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 ROT ,. 3 3 $ 1 0 0 0    NB. rename from RR
glPolygonMode GL_FRONT, GL_FILL  NB. Front and Back: Full Paint
glPolygonMode GL_BACK, GL_POINT  NB. Back: Point(Hidden)

glBegin GL_QUADS

(>D_COL) face_col (0.6&*) L:0 D_P NB. reduced 0.6 size

(>L_COL) face_col (0.6&*) L:0 L_P

(>U_COL) face_col (0.6&*) L:0 U_P

(>R_COL) face_col (0.6&*) L:0 R_P

(>B_COL) face_col (0.6&*) L:0 B_P

(>F_COL) face_col (0.6&*) L:0 F_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
glLineWidth 0.5
```

```
glColor 0 0 0
glBegin GL_POLYGON
 glVertex y.
glEnd ''
)


draw_frames =: 3 : 0
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glTranslate 0 0 _1
glRotate ROT ,. 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 F_P
face_frame > (0.6&*) L:0 U_P
face_frame > (0.6&*) L:0 R_P
face_frame > (0.6&*) L:0 L_P
face_frame > (0.6&*) L:0 B_P
face_frame > (0.6&*) L:0 D_P
)


tc =: 3 : 0
:
'I J R' =. x.
'X Y' =. y.
0 < ((%:2)*R) - (%: +/ *: (X-I), (Y-J))
)


NB. project the picture on the screen ==================
aa_g_size =: verb define
wh =. glqwh ''
glViewport 0 0, wh
glMatrixMode GL_PROJECTION
glLoadIdentity ''
glOrtho _2.5 2.5 _2.5 2.5 _2.5 2.5
NB. gluPerspective 60, (%/wh), 1 30
)


NB. べき乗を積の繰り返しへ  ========================================
require 'system¥main¥regex.ijs'
NB.
```

```
NB.     calc '(ER)3R(eR)3R'    2012/1/20
NB. ERERERReReReRR

calc =: 3 : 0
y =. y. -. ' '
while. '(' e. y do. y =. par2mul y end.
pow2mul y
)
NB.     calc 'Fr2'
NB.     calc 'ab3(xy2z)3d2'
NB. abbbxyyzxyyzxyyzdd

NB. power to multiple - revised 2011/12/13
NB.    pow2mul 'ab3xy2zxy2zxy2zd2'
NB. abbbxyyzxyyzxyyzdd

pow2mul =: 3 : 0
NB. single alpha to alpha + 1 - revised 2011/12/13
p =. y. , ({: y.)          NB. revised 12/14
PP =. 2 <¥ p
PPP =. ; ad1 L:0 PP
NB. 'ab2c3' => 'abbccc'
P =. '[[:alpha:]]+[[:digit:]]' rxall PPP
P1 =. _1{ L:0 P
P2 =. _2{ L:0 P
P1 p2m P2
)

asc =: a.&i.

alp =: (64&< * <&123)@asc
num =: (47&< * <&58)@asc

ad1 =: 3 : 0
'A1 B1' =. y.
AD1 =. y.
if. alp B1 do. AD1 =. A1, '1' end.
if. num A1 do. AD1 =. '' end.
AD1
)


p2m =: 3 : 0
:
n =. #y.
```

```
i =. 0
PM =. ''
while. i < n
  do.
    PM =. PM , (". >i{x.) # (>i{y.)
    i =. i + 1
  end.
PM
)

NB. かっこをはずす 2011/12/14 ============================================


NB.    par2mul 'ab3(xy2z)3d2'
NB. ab3xy2zxy2zxy2zd2
par2mul =: 3 : 0
P0 =. '.*(¥(.+¥)).*' rxmatches y.  NB. $P=1 2 2
P2 =. ,{:"2 P0                     NB. take 2nd row of P
P3 =. ({. P2) + i. {: P2
Q0 =. P3 { y.
A =. ({. P2) {. y.
B =. (>: >:{: P3) }. y.
C =. (>: {: P3) { y.
Q1 =. }. }: Q0
Q =. ,(". C)#,: Q1
A, Q, B
)

NB.    unpar 'ab3(xy2z)3d2(p2q)2e3'
NB. ab3xy2zxy2zxy2zd2p2qp2qe3
unpar =: 3 : 0
np =. +/ '(' = y.
i =. 0
R =. y.
while. i < np
  do.
    R =. par2mul R
    i =. i + 1
  end.
R
)


NB. Alpha(Large) + underbar(_) => alpha(small)
val=: a. & i.
```

```
to_small =: (32&+) &. val
to_large =: (-&32) &. val

NB. eg. und2small 'AB_CDE_F' => 'AbCDeF'
und2small =: 3 : 0
Y =. '[[:alpha:]]¥_' (to_small@{.) rxapply y.
)



NB. Joyner's Notation ============================================
J_DA =: (9&#) L:0 'L';'F';'R';'B';'U';'D'

NB. Joyner to Shimanouchi
Joy2Sim =: 3 : 0
Y1 =: und2small y.
Y2 =: pow2mul unpar Y1
Y3 =. J2S Y2
)


J2S =: 3 : 0"(0)
select. y.
  case. 'F' do. 's'
  case. 'R' do. 'e'
  case. 'L' do. 'w'
  case. 'B' do. 'n'
  case. 'U' do. 't'
  case. 'D' do. 'b'
  case. 'S' do. 'i'
  case. 'M' do. 'j'
  case. 'E' do. 'k'
  case. 'f' do. 'S'
  case. 'r' do. 'E'
  case. 'l' do. 'W'
  case. 'b' do. 'N'
  case. 'u' do. 'T'
  case. 'd' do. 'B'
  case. 's' do. 'I'
  case. 'm' do. 'J'
  case. 'e' do. 'K'
end.
)

Sim2Joy =: 3 : ', S2J y.'
```

```
S2J =: 3 : 0"(0)
select. y.
  case. 'T' do. 'u'
  case. 't' do. 'U'
  case. 'N' do. 'b'
  case. 'n' do. 'B'
  case. 'B' do. 'd'
  case. 'b' do. 'D'
  case. 'S' do. 'f'
  case. 's' do. 'F'
  case. 'E' do. 'r'
  case. 'e' do. 'R'
  case. 'W' do. 'l'
  case. 'w' do. 'L'
end.
)

val=: a. & i.
chr=: val ^:_1

to_large =: (chr@(val - (32"_)))
to_small =: (chr@(val + (32"_)))

NB. e.g.   display ". (rubman 'FE'), ' ALL_COJ'
NB. Rubik's Maneauver p.25
NB. for display
rubman =: 3 : 0
'p q' =. y.
P =. p, p
Q =. q, q
M1 =. P,' & ', ,3 9$P,' & ',Q,' & '
M2 =. P,' & ', ,3 9$P,' & ',(to_small Q),' & '
M2, ' & ', M1
)

NB. copy the output by mouse, then paste it on the edit box
NB. e.g. rubm 'RE' => 'E R E R E R R e R e R e R R'
NB. input onto the edit box, "rubm 'RE' CR "
rubm =: 3 : 0
'P Q' =. y.
M1 =: (,3 4$Q,' ',P,' '), P
M2 =: (,3 4$(to_small Q),' ',P,' '), P
M1, ' ', M2
)
```

```
rubc =: 3 : 0
'P Q' =. y.
D1 =: calc '(',Q,'2',P,'2)3',P,'2'
D2 =: calc '(',(to_small Q),'2',P,'2)3',P,'2'
NB. C1 =: calc '(E2F2)3F2'
NB. C2 =: calc '(e2F2)3F2'
D12 =: D1, D2
_2 }. , (14 2$D12), '"(1) ' & '
)



NB. display revised Joyner's notation in number
NB. revised 2012/3/1
NB. e.g. RUBIK_DISPLAY ''

NB. test Rubik Maneuver sample cycle
N_TRMAN =: (14 21;23 30) C. i. 54
NB. display Rubik Maneuver in number => RUBIK_DISPLAY N_TRMAN

RUBIK_DISPLAY =: 3 : 0
RUB0 =. 7 10$' '
if. 0 = # y.
 do.  NB. Orignal Index
  RUB =. <"(0) 3 3$i.9
  RUB1 =. '': 2'': L:0 RUB
  RUB2 =. '': 2'': L:(0) 9 + L:0 RUB
  RUB3 =. '': 18 + L:0 RUB
  RUB4 =. '': 27 + L:0 RUB
  RUB5 =. '': 36 + L:0 RUB
  RUB6 =. '': 45 + L:0 RUB
 else. NB. Index from argument y. e.g. Rubik Maneuver test
  RR =: (6, 9)$ y.
  RRA =: <"(1) RR
  'RUB1 RUB2 RUB3 RUB4 RUB5 RUB6' =: (3, 3) $ L:1 <"(0) L:0 RRA
  RUB1 =. '': 2'': L:0 RUB1
  RUB2 =. '': 2'': L:0 RUB2
  RUB3 =. '': 2'': L:0 RUB3
  RUB4 =. '': 2'': L:0 RUB4
  RUB5 =. '': 2'': L:0 RUB5
  RUB6 =. '': 2'': L:0 RUB6
end.
RUB11 =. RUB0,'"(1) RUB1,'"(1) RUB0,'"(1) RUB0
RUB22 =. RUB2,'"(1) RUB3,'"(1) RUB4,'"(1) RUB5
RUB33 =. RUB0,'"(1) RUB6,'"(1) RUB0,'"(1) RUB0
```

RUB11, RUB22, RUB33
)

NB. RDJ =: ' U L F R B D '
NB. display RDJ

```
NB. Several Idioms from 「宝島社」攻略本 ==================================

NB. 上段コーナー・キューブへの移動 p.13

cmov =: 3 : 0  NB. revised x. argument no. 2012/2/22
:
R =. 1}y.
r =. to_small R
F =. 0}y.
f =. to_small F
select. x.
 case.  1 do. R,f,r,F                    NB. p.13A
 case. _1 do. f,R,F,r                    NB. p.13B
 case.  3 do. R,f,r,F, R,f,r,F, R,f,r,F  NB. p.13C
 case.  2 do. R,f,r,F, R,f,r,F           NB. p.13D twist-cw of corner cubelet
 case. _2 do. f,R,F,r, f,R,F,r           NB. p.13E twist-ccw of corner
cubelet
end.
return.
if. 0 = #y.
 do.
  select. x.
   case.  1 do. 'RfrF'             NB. p.13A
   case.  2 do. 'fRFr'             NB. p.13B
   case.  3 do. 'RfrF RfrF RfrF'   NB. p.13C
   case.  4 do. 'RfrF RfrF'        NB. p.13D
   case.  5 do. 'fRFr fRFr'        NB. p.13E
  end.
 else.
  select. x.    NB. e.g. 3 cmov 'BL' => 'LblB LblB'
   case.  1 do.  (1}y.), (to_small 0}y.), (to_small 1}y.), (0}y.)
NB. p.13A
   case.  2 do.  (to_small 0}y.), (1}y.), (0}y.), (to_small 1}y.)
NB. p.13B
   case.  3 do.  ya, ya, ya =. (1}y.), (to_small 0}y.), (to_small 1}y.),
(0}y.)  NB. p.13C
   case.  4 do.  ya, ya =. (1}y.), (to_small 0}y.), (to_small 1}y.), (0}y.)
NB. p.13D
   case.  5 do.  yb, yb =. (to_small 0}y.), (1}y.), (0}y.), (to_small 1}y.)
NB. p.13E
  end.
end.
)

NB. 上段エッジ・キューブの移動 p.14, 15
```

```
NB. e.g. 1 emov 'R', 2 emove 'F', 3 emov 'L'
emov =: 3 : 0  NB. revised 2012/2/5
:
if. 1 = ColInp do. y =. col2fac y. else. y =. y. end.
select. y
  case. 'R' do.
    select. x. case. 1 do. z =. 'RsrS'      NB. p.14B rotate clockwise 90 deg
               case. 2 do. z =. 'RRsrrS'    NB. p.14A rotate clockwise 180 deg
               case. 3 do. z =. 'RRRsrrrS'  NB. p.14C rotate clockwise 270 deg
               case. 4 do. z =. 'sRSr'      NB. p.14D
               case. 5 do. z =. 'RErrEERE'  NB. p.14E and Revised
    end.
  case. 'B' do.
    select. x. case. 1 do. z =. 'BMbm'
               case. 2 do. z =. 'BBMbbm'
               case. 3 do. z =. 'BBBMbbbm'
               case. 4 do. z =. 'MBmb'
               case. 5 do. z =. 'BEbbEEBE'
    end.
  case. 'L' do.
    select. x. case. 1 do. z =. 'LSls'
               case. 2 do. z =. 'LLSlls'
               case. 3 do. z =. 'LLLSllls'
               case. 4 do. z =. 'SLsl'
               case. 5 do. z =. 'LEllEELE'
    end.
  case. 'F' do.
    select. x. case. 1 do. z =. 'FmfM'
               case. 2 do. z =. 'FFmffM'
               case. 3 do. z =. 'FFFmfffM'
               case. 4 do. z =. 'mFMf'
               case. 5 do. z =. 'FEffEEFE'
    end.
end.
if. 1 = ColInp do. z =. colx z end.
z
)

NB. e.g. 2 emov '' => 'RRsrrS'
emov0 =: 3 : 0
:
if. 0 = #y.
 do.
NB.
 else.
```

```
  'R s r S' =. (0}y.), (to_small 1}y.), (to_small 0}y.), (1}y.)
end.
select. x.
   case.  1 do. 'RsrS'
   case.  2 do. 'RRsrrS'
   case.  3 do. 'RRRsrrrS'
   case. _1 do. 'rsRS'
end.
)

NB. 下段コーナー・キューブの位置の交換 p.17
NB. cpos
cpos =: 3 : 0
if. 1 = ColInp do. y =. col2fac y. else. y =. y. end.
select. y
  case. 'F' do. z =. 'bDBrBRbDD'
  case. 'R' do. z =. 'lDLbLBlDD'
  case. 'B' do. z =. 'fDFlFLfDD'
  case. 'L' do. z =. 'rDRfRFrDD'
end.
if. 1 = ColInp do. z =. colx z end.
z
)

NB. 下段コーナー・キューブの向きの交換 p.19
cdir =: 3 : 0
if. 1 = ColInp do. y =. col2fac y. else. y =. y. end.
z =. (to_small y), 'DD', (y), 'D',(to_small y), 'D', (y), 'DD'
NB. 'bDDBDbDBDD'
if. 1 = ColInp do. z =. colx z end.
)

NB. 下段のエッジ部分に中段からエッジ・キューブを移動する p.21
dfrch =: 3 : 0
:
if. 1 = ColInp do. y =. col2fac y. else. y =. y. end.
select. x.
  case.  1 do. z =. (y), 'e', (to_small y), 'E', (to_small y), 'e', (y)
  case. _1 do. z =. (to_small y), 'E', (y), 'e', (y), 'E', (to_small y)
  case.  2 do.
 z =. (y),'e',(to_small y),(to_small y),'e',(y),'e ',(to_small y),'E',
(y),'e',(y),'E',(to_small y)
end.
NB.   case.  1 do. 'RerEreR'          NB. from BR-edge to DR-edge
NB.   case. _1 do. 'rEReREr'          NB. from FR-edge to DR-edge
```

```
NB.  case.  2 do. 'RerreRe rEReREr'  NB. exchange RD-edge to DR-edge
NB. end.
if. 1 = ColInp do. z =. colx z end.
)

NB. 中段エッジ・キューブの位置の交換 modified of p.23
epos =: 3 : 0
if. 1 = ColInp do. y =. col2fac y. else. y =. y. end.
z =. y, y, 'e', y, y, 'E'
NB. 'FFeFFE'
if. 1 = ColInp do. z =. colx z end.
)

NB. 上段または下段エッジ・キューブの巡回置換 Joyner p.355の手順1
NB. e.g. ecycle 'U', or, ecycle 'D'
ecycle =: 3 : 0
NB. 1 ecycle y.
if. 1 = ColInp do. y =. col2fac y. else. y =. y. end.
x =. 'D'
NB. select. x.
NB.   case. 1 do. x =. 'D'
NB.   case. 2 do. x =. 'U'  NB. need revised !!
NB. end.
select. y
    case. 'F' do. z =. 'mm', (to_small x), 'M', x, x, 'm', (to_small x), 'mm'
    case. 'B' do. z =. 'MM', (to_small x), 'm', x, x, 'M', (to_small x), 'MM'
    case. 'R' do. z =. 'ss', (to_small x), 'S', x, x, 's', (to_small x), 'ss'
    case. 'L' do. z =. 'SS', (to_small x), 's', x, x, 'S', (to_small x), 'SS'
end.
if. 1 = ColInp do. z =. colx z end.
)

NB. 上段または下段エッジ・キューブのクロス交換 Joyner p.355の手順7
NB. e.g. ecross 'U'(上段), ecross 'D'(下段)
ecross =: 3 : 0
z =. calc 'F2L2', (y.) ,'2(F2L2)3', (y.) ,'2L2F2'
)

NB. Joyner p.92-93, p.355 / swap two pairs of edges
=========================================
eswap2 =: 3 : 0
'p q' =: y.
z =. p, p, q, q, p, p, q, q, p, p, q, q
)
```

```
NB. Generate Rubik Problem by Rundom Numbers ======================
NB. e.g. rnd 8 => 'gkRWOrGw'
rnd =: 3 : 0
RND =. 'uUlLfFrRbBdD'
if. 1 = ColInp do. RND =. 'bBkKrRgGoOwW' end.
(y. ? 12) { RND
)

NB. Several Initial Settings ====================================
NB. run 10, 1 for start
RecDA1 =: 'gkRWOrGw'
RecDA2 =: RecDA1, 'wkbOkKw'
RecDA3 =: RecDA2, 'RkrK'
```