# ルービック・キューブの楽しみ方いろいろ
## －手で動かす楽しみと群論・Ｊ立体グラフィックス－

西川　利男

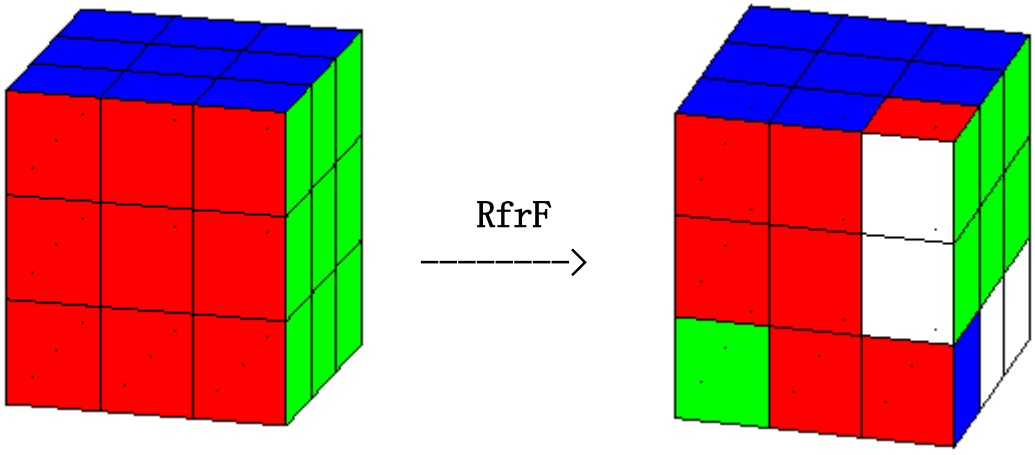
ルービック・キューブには、実際に手に持って動かして遊ぶ楽しみがあり、同時に群論の応用としての解析、さらには J-OpenGL による立体グラフィックスのダイナミックなシミュレーションなど、いろいろな楽しみ方がある。

先日、志村正人氏から私宛にルービック・キューブのＪプログラムの問い合わせが来たということで、すでに数年前、JAPLA 研究会で、数回にわたって発表した私のレポートをあらためて精査してみた。

今回は、まずルービック・キューブを動かす楽しみを第一の視点として易しく紹介し、群論による解析やＪグラフィックスの詳細は、先のレポートを随時参照していただくようにした。

## １．ルービック・キューブを動かしてみよう

ルービック・キューブは３ ｘ ３ ｘ ３の色分けされた小片（Cubelet などと呼ぶ）から成る一種のサイコロで遊ぶパズル・ゲームである。面と回転操作の名前及び色わけの仕方にはいろいろとあるが、ここでは日本での解説書などで標準とされている方式（かつての国際標準）を用いた。 たとえば、前(赤)F 面の右下のコーナーキューブを右上に移動する操作の記述は以下のようにする。




Rubik Cube の面の名前　上(青)U, 前(赤)F, (緑)R, 左(黄)L, 後(橙)B, 下(白)D
Rubik Cube の操作　…　面に対応して名付けられている。
コーナキューブの回転　上面での反時計 u, 時計 U、前面での反時計 f, 時計 F
右面での反時計 r, 時計 R、左面での反時計 l, 時計 L
後面での反時計 b, 時計 B、下面での反時計 d, 時計 D
エッジキューブの回転　中央(middle) 左面から見て反時計 m, 時計 M、
側面(side)　前面から見て反時計 s, 時計 S、

横面(equator) 上面から見て反時計 e, 時計E

## ２．ルービック・キューブー揃え方の基本

以前に発表したレポートから、引用する。

### 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、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。

実際のルービック・キューブでは、それなりの定石があるが、いろいろ試しつつ、やってみるしかない。簡単に行かないことが楽しみにもなるのだろう。

## ３．ルービック・キューブと群論そしてＪ

ルービック・キューブの操作の攻略法はだれでも究めたいところである。
ところが、これがなかなか難しい！　数学としては群論の助けを借りることになる。

ここでも、以前のレポートから採録する。

### 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。

ルービック・キューブの解析の難しさのポイントは次の点にある。

ルービック・キューブの各 cubelet（位置、色）の移動は、本来
３次元空間内における運動（x(t), y(t), z(t)）である。

↓

これを、文字、記号の置き換えとして計算する。

つまり群論の置換群である。この考え方が、同型写像(Isomorphism)と呼ばれ、群論が数学の他の分野と違った価値観を持ち、有能な理由である。

**Jと群論**

さらに、われわれにとって、有利なことは、群論の置換操作はJのプリミティブの動詞C.として備えられている。なお、群の元つまり操作とはJと同じく動詞である。

**ルービック・キューブとJ**

つまるところ、ルービック・キューブの動きは、Jでプログラムすることが出来る、ということになる。加えて、JのOpenGL機能により３次元のグラフィックス・システムが可能になる。

## ４．J602によるルービック・キューブのプログラム

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は最初はJ3で始めたが次にJ-OpenGLによりJ4で開発した。最後にJ6版にも移植して、動くことを確かめた。

J6では、利用の便利さからつぎのようにUSB上にプログラムを格納している。

¥j602¥Open System¥j602-User¥user¥OpGLN_RubikJ6.ijs

プログラムを実行するには

run ''

とすると、宝島社、「頭を鍛える」版の色設定で実行される。

プログラムでは、ルービック・キューブの単独の操作だけでなく、一連の連続操作をプログラムとしてあらかじめ入力した後、一括操作する機能を持たせた。さらに、有名な「ルービック・マヌーバ」などの定石も行えるようにした。

以下にプログラム実行のいろいろなオプション引数を示す。

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

**ルービック・キューブのJプログラム操作のいろいろな方法**

最初にあげた、ルービック操作コマンドの文字を用いるが、いろいろな方法がある。

(0) 直接入力実行

例えばR f r Fのように入力すると、ただちに回転、実行され、表示が変わる。

このとき、ルービック・キューブ全体はいろいろ視点を変えて、観察できる。

X, x => 左右、 Y, y => 上下、 Z, z => 反時計回転、時計回転

(1) 入力窓(Input Edit Box)への文字列入力

(1.1) 例えばR f r F(CR)のように文字列入力した後
Autoボタンのクリックで、一括でルービック操作
Stepボタンのクリックで、１ステップずつルービック操作

なお、実際の動きはキューブを**rbx**として、次のように操作されることに注意。
F (r (f (R **rbx**) ) ) … 関数として、右から順次実行

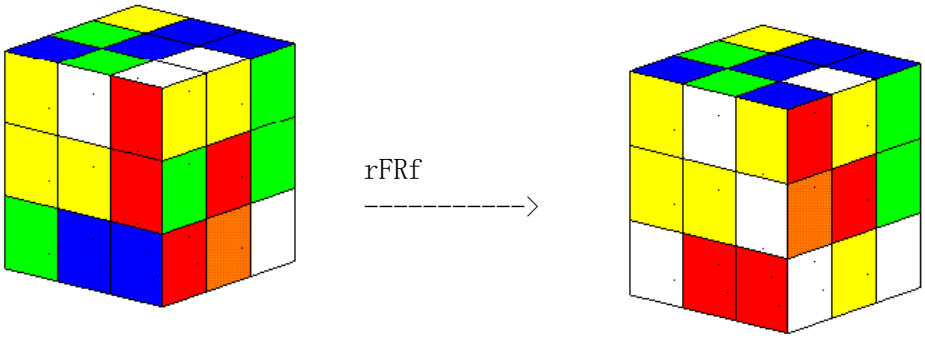
(1.2) calc 'F2 L2 U2 (F2 L2)3 U2 F2'(CR) のように文字列入力した後
Compileボタンを押すと、繰り返し、かっこが展開されるので
FF LL UU FF LL FF LL FF LL UU LL FF さらに(CR)を打って
後はAuto, Stepボタンで操作を行う。

(1.3) 定石「ルービック・マヌーバ」などは

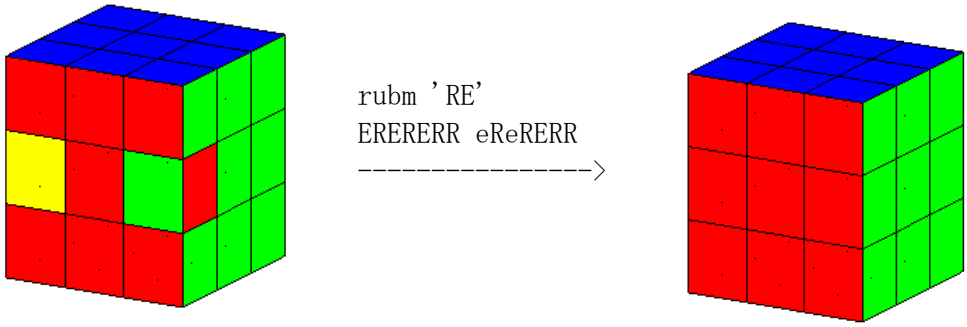
rubm 'FE'、 rubm 'RE'、 rubm 'FM' などと入力して行える。
Joyner の本、宝島社の本からいろいろな定石プログラムを定義してある。

## ５．OpenGL による 3D グラフィックスの実際

ルービック・キューブの操作を 3D グラフィックスの実行の実際を示す。
いろいろ動かして、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、定石 rFRf を使って移動させると、右図のようになる。




上段が揃ったら攻略法として、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。ここでルービック・マヌーバと呼ばれるすばらしい定石の効果を示す。最後に残った２つのキューブレットの位置を変えず向きだけを変える。




## 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).
[5] 西川利男「技術数学に向けての群論のすすめ－ルービック・キューブで学ぶ群論と 3D コンピュータ・シミュレーション－」日本技術史教育学会、2012 年度総会.
[6] 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

ールービック・キューブのシミュレーション」北海道情報大学紀要 10, 11 (1998).

## ルービック・キューブの J602 プログラム・リスト
## 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_RubikJ6.ijs  2012/3/24
NB. J602 version imported from J402 2012/3/20
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
```

```
NB. J6 version ============================================================
wr =: 1!:2&2
require 'opengl gl3'
coinsert 'jgl3'

A=: 0 : 0
pc a closeok;pn "Rubik Cube Simulated";
menupop "&Help";
menu help "&Help" "" "" "";
menupopz;
xywh 6 22 200 200;cc g isigraph opengl 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 Fo□  p 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   R}Hh F□ p 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=: a_run
a_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 A
NB. glaRC ''
ogl =: '' conew 'jzopengl'       NB. for J6
ROT =: 0 0 0  NB. rename from RR
Key =: ''
NL =: 0       NB. NL(New Line) Flag = 0
NB. for J6, modify use g_draw_init
NB. glaFont 'arial 30'
NB. 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. ftHg□ËubvFپ
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. @Fپ
      U_COL =: 'WWWWWWWWW'
      L_COL =: 'RRRRRRRRR'
      F_COL =: 'YYYYYYYYY'
      R_COL =: 'OOOOOOOOO'
      B_COL =: 'GGGGGGGGG'
      D_COL =: 'BBBBBBBBB'
    case. 1  do. NB. □ĖubvFپ
      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 ========================================
a_g_paint =: verb define
RC =: rc__ogl ''
if. RC do. g_draw_init wh_ogl end.
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT
g_size ''
g_draw_rubik ''
g_draw_frames ''
NB. drawface1 texdraw pat_d1  NB. 2012/3/6
g_drawtext''
NB. glaSwapBuffers ''
show__ogl ''
)


NB. key-in x, y, z, X, Y, Z for rotation ================
a_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)
a_g_paint ''
)

g_draw_init =: 3 : 0
glViewport 0 0, y
('arial';30) glaUseFontBimaps__ogl 32 95 32
glMatrixMode GL_PROJECTION
glLoadIdentity ''
glOrtho _2.5 2.5 _2.5 2.5 _2.5 2.5
)

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
l1 =: l0 & C.        NB. l1 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 =: ssˆ:_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 ============
g_drawtext =: verb define
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glColor 0 0 0 0
glRasterPos _2.0 _2.0 0
glaCallLists 'Test'
NB. if NL = 1 with Key = '-', write two lines 2012/1/31
if. NL = 0
  do.
     glRasterPos _2.3 2.1 0
     glaCallLists Key     NB. for J6
  else.
     lin2mul Key  NB. revised multilines 2012/2/1
end.
glRasterPos _2.0 _2.4 0
glaCallLists (5 ": ROT)  NB. for J6, 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
    glaCallLists LIA
    LIB =. ii}. LIN
    LIN =. LIB
    jj =. jj + 1
end.
glRasterPos _2.3, (2.1-0.3*jj), 0
glaCallLists LIB
)

a_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'
)

a_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

g_draw_rubik ''
g_draw_frames ''
Key =: ''
g_drawtext ''
istep =: 0
show__ogl ''
NB. glaSwapBuffers ''
wd 'set Program ""'
iPDA =: ''
wd 'setfocus g'
)
```

```
a_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.
a_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'
)

a_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'
)

a_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! ====
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)
)

boundTex =: 3 : 0    NB. 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)
)
NB. Bound =:  >, boundTex L:0 Point  NB. for Texture 2012/3/6
Bound =:  >, boundary L:0 Point
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 ================================================


NB. test_COL =:
'RED';'RED';'GREEN';'GREEN';'GREEN';'GREEN';'BLUE';'RED';'BLUE'

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


polygon=: 4 : 0
glColor4d 4{.x, 1     NB. for J6
NB. glColor x
glBegin GL_POLYGON
 glVertex y
glEnd ''
)

g_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
glColor4d 0 0 0 1
NB. glColor 0 0 0
glBegin GL_POLYGON
 glVertex y
glEnd ''
)

fsymbol =: 3 : 0
glLineWidth 1.5
glColor 0 0 0
XS =. 0.6 * 0 { > 0 { 8 { F_P
YS =. 0.6 * 1 { > 0 { 8 { F_P
ZS =. 0.6 * 2 { > 0 { 8 { F_P
glBegin GL_LINE_LOOP
  glVertex  (XS+0.2), (YS-0.4), ZS
  glVertex  (XS+0.3), (YS-0.2), ZS
  glVertex  (XS+0.4), (YS-0.4), ZS
glEnd ''
)

g_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
NB. fsymbol ''
)



NB. project the picture on the screen ====================
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. ○ςJdgg@=========================================
NB. require 'system¥main¥regex.ijs'
require 'regex'
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. ~ 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 u□Ë vU{ ==================================

NB. iR[i[EL[uuL 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. iGbWEL[wü ڑ 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. iR[i[EL[ɯ ʒɯ̆  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. iR[i[EL[ɯ̈  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. ĩGbWɒiGbWEL[u ɻ 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. iGbWEL[ɯ ʒɯ̆  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. i ¸ GbWEL[ù u Joyner p.355 菇 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. i ¸ GbWEL[ũNX Joyner p.355 菇 7
NB. e.g. ecross 'U'(i), ecross 'D'(i)
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'



NB. Get X, Y of Left_Mouse Down
a_g_mbldown=: 3 : 0
d=. ". sysdata
m_x=:(0{d) * 1000 % (2{d)  NB. X value
m_y=:(1{d) * 1000 % (3{d)  NB. Y value
)
```

```
NB. j4 versiopn ============================================
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'
```