

## J-OpenGL による 4 進フラクタル立体木のグラフィックス

西川 利男

今年の大学センター試験の出題をきっかけとして、3 進法およびそれを活用した 3 進フラクタル木の J プログラムについて先に報告した [1][2]。図形表示の値を 3 進法で表すことで、フラクタル木のグラフィックスが、ごく自然に行われた。

それでは 4 進法ではどうだろうか？同じ発想を展開すると、4 進法では、3 次元の立体空間内で均等に枝が伸びるフラクタルの木が生成するはずである。

しかし、ここで一つの難問が発生する。3 次元立体空間内の図形をどう表示するかである。これには OpenGL グラフィックスで行う以外はない！これまで何回か紹介した [3][4][5] もの、私としては、まだ OpenGL の入り口をのぞいたにすぎない。期せずして、4 進法フラクタルの木により、OpenGL の手法を進める良きテーマになった。

### 1. J での 4 進法の処理

自然数の 10 進法表示から 4 進法表示への変換は J ではつぎのプリミティブ(#.) を使って、簡単に行われる。

```
qfd =: 3 : '4 #. ^:(_1) y.'
```

たとえば、次のように変換される。

```
N =: <"(0) i. 16
```

```
Q =: <"(1) qfd i. 16
```

```
N ,: Q
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 0	0 1	0 2	0 3	1 0	1 1	1 2	1 3	2 0	2 1	2 2	2 3	3 0	3 1	3 2	3 3

### 2. 4 進数の 3 次元空間内での位置の座標値

3 次元座標値を (x, y, z) とするとき、4 進数の座標値は次のような値となる。

0: (1, 0, 0)

1: (cos θ, sin θ, 0)

2: (cos θ, -sin θ, 2/√3 sin θ)

3: (cos θ, -sin θ, -2/√3 sin θ)

where θ = 109.47°

これは化学構造において、メタン(CH<sub>4</sub>)のC原子を原点(0, 0, 0)とし、CとHとの結合距離(1.54Å)を1としたときの4つのH原子の位置に相当する。また角度 θ は結合の2面体角(dihedral angle)と呼ばれる。

### 3. 4進数の3次元空間内の座標値の計算

4進数の桁数はフラクタル図形の次数に対応する。

4進数の桁数(フラクタルの次数)を指定したときに4進数の組は次の quart により得られる。

```
quart =: 3 : 0
qufd i. 4^y.
)
```

```
quart 1
0
1
2
3
```

```
quart 2
0 0
0 1
0 2
0 3
1 0
1 1
1 2
1 3
2 0
2 1
2 2
2 3
3 0
3 1
3 2
3 3
```

さらに、これを用いて、空間内の位置の座標値はプログラム qtopos のより計算される。プログラムは後述。

```
qtopos 1
+-----+
| 1 0 0 |
+-----+
|_0. 333313 0. 942816 0 |
+-----+
|_0. 333313 _0. 942816 0. 816503 |
+-----+
|_0. 333313 _0. 942816 _0. 816503|
+-----+
```

```

qtopos 2
+-----+
| 1.5 0 0 |
+-----+
| 0.833343 0.471408 0 |
+-----+
| 0.833343 _0.471408 0.408251 |
+-----+
| 0.833343 _0.471408 _0.408251 |
+-----+
| 0.166687 0.942816 0 |
+-----+
| _0.49997 1.41422 0 |
+-----+
| _0.49997 0.471408 0.408251 |
+-----+
| _0.49997 0.471408 _0.408251 |
+-----+
| 0.166687 _0.942816 0.816503 |
+-----+
| _0.49997 _0.471408 0.816503 |
+-----+
| _0.49997 _1.41422 1.22475 |
+-----+
| _0.49997 _1.41422 0.408251 |
+-----+
| 0.166687 _0.942816 _0.816503 |
+-----+
| _0.49997 _0.471408 _0.816503 |
+-----+
| _0.49997 _1.41422 _0.408251 |
+-----+
| _0.49997 _1.41422 _1.22475 |
+-----+

```

フラクタルの次数(=4進数の桁数)が上がるごとに、木の枝は係数 0.5 を掛けた値だけ延長される。上の計算値はこのような(x, y z)の座標値を示している。すなわちこれによりフラクタル木が描画されることになる。

#### 4. JのOpenGLプログラム

OpenGLとは本来C++言語向けの仕様で、立体の描画を含めた汎用のグラフィックス処理のプログラム書法である。

JではこれをJ向きに書式をあわせたライブラリ g13としてサポートしている。グラフィックス操作はgl… やglu…のようにOpenGLと同じコマンドで行われる。

つぎのような構成を基本としている。

- グラフィックスのためのウィンドウズ・フォームの作成

```
A =: noun define
pc a ... ;
xywh ... ;
cc g isigraph ... ;
)
```

- プログラムの実行=フォームの実行

```
a =: verb define
wd A
glaRC '' (OpenGLのdllを起動)
(いろいろな初期設定やコンパイルなど)
wd 'pshow; ptop'
)
```

つぎのプログラムは起動とともに、イベントドリブンとして同時に実行される。

- 画面の表示

```
a_g_paint =: verb define
glClearColor ...
glTranslate ...
glRotate ...
...
)
```

- 画面の大きさなど

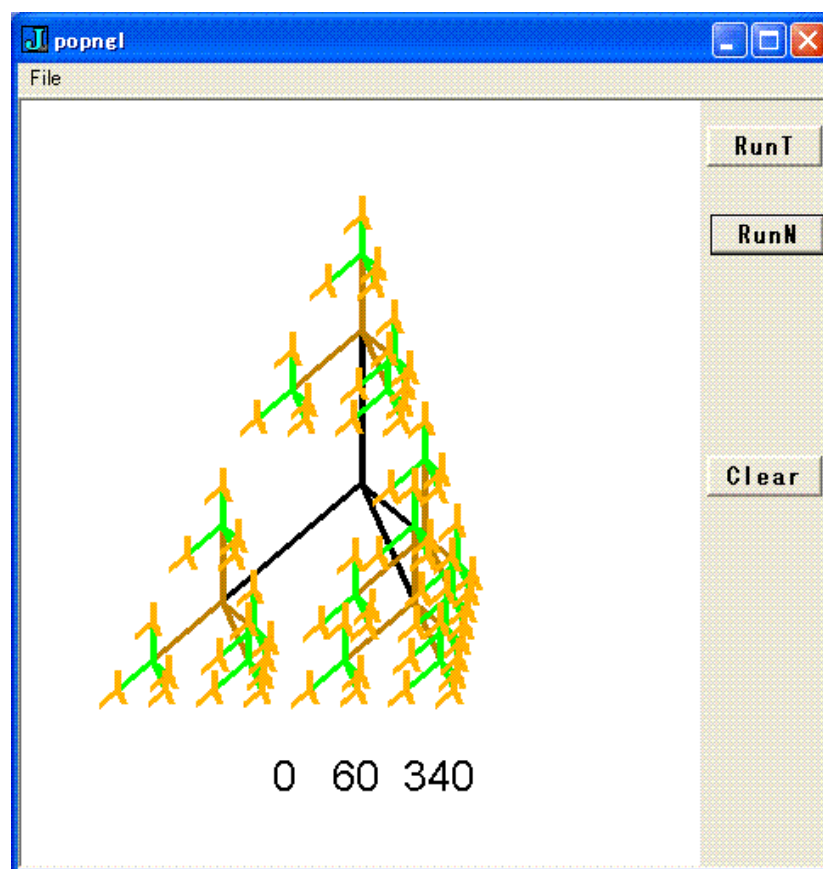
```
a_g_size =: verb define
...
)
```

- 画面上でのキー入力

```
a_g_char =: verb define
...
)
```

実際のプログラムは稿末を参照のこと。また、多くの細かい指定の入ったプログラムとなっているので、OOPのクラスプログラムとし、単なる実行には簡単なプログラムだけで行えるようにした。

## 5. 4進数のフラクタル木のグラフィックスの実際



上のようなウィンドウ画面上で、ボタン「RunN」を押すごとにフラクタルの次数が1ずつ上がり4進の木が大きく成長するようすが観察される。

OpenGLの機能として、キーボードからx, X, y, Y, z, Zを打つことで、各軸での回転により、いろいろ視線を変えて立体として見られる。また、立体図形の大小 b, s 移動、左右 l, r 上下 u, d 遠近 f, n も可能である。

- [1] 西川利男「Jによる3進法の処理—その1、3進法の基本と大学センター試験の問題を解く」JAPLA研究会資料 2013/3/23
- [2] 西川利男「Jによる3進法の処理—その2、3進数によるフラクタル・グラフィックス、OOPプログラムの例として」JAPLA研究会資料 2013/3/23
- [3] 西川利男「Jのg13-OpenGLによるグラフィックス—その1」JAPLA研究会資料 2009/9/26
- [4] 西川利男「Jのg13-OpenGLによるグラフィックス—その2—正8面体と正12面体を動かす—」JAPLA研究会資料 2009/9/26
- [5] 西川利男「JのOpenGLグラフィックス—その3—J602版OpenGL / サイコロの回転(cube)を例として—」JAPLA研究会資料 2009/10/24



NB. tree4\_OOP\_OPGL.ijs  
 NB. 4 進の木  
 NB. Run OpenGL with Object Oriented Method  
 NB. 2013/4/3

wr =: 1!:2&2

require 'trig'

COS =: 3 : 'cos rfd y.'

SIN =: 3 : 'sin rfd y.'

run =: 3 : 0

Path =: 1!:40 ''

load Path, 'user\classes\popngltr4.ijs'

object =: '' conew 'popngltr4'

)

tfd =: 3 : '3 #. ^:(\_1) y.'

dft =: 3 : '3 #. y.'

NB. quaternary tree =====

qfd =: 3 : '4 #. ^:(\_1) y.'

dfq =: 3 : '4 #. y.'

TH =: 109.47

OR =: 0, 0, 0

PQ =: (1, 0, 0),: (COS TH), (SIN TH), 0

PQ =: PQ, (COS TH), (- SIN TH), (-: %:3)\*SIN TH

PQ =: PQ, (COS TH), (- SIN TH), (- -: %:3)\*SIN TH

PD =: OR, PQ NB. Vertex Values

CD =: (0, 1);(0, 2);(0, 3);(0, 4) NB. Connect Parameters

CT =: (1, 2);(1, 3);(1, 4);(2, 3);(2, 4);(3, 4)

```

coclass 'popngltr4'

require 'trig'

COS =: 3 : 'cos rfd y.'
SIN =: 3 : 'sin rfd y.'

NB. 4-th tree vertex =====
TH =: 109.47
OR =: 0, 0, 0
PQ =: (1, 0, 0),: (COS TH), (SIN TH), 0
PQ =: PQ, (COS TH), (- SIN TH), (-: %:3)*SIN TH
PQ =: PQ, (COS TH), (- SIN TH), (- -: %:3)*SIN TH
PD =: OR, PQ
CD =: (0, 1);(0, 2);(0, 3);(0, 4)
CT =: (1, 2);(1, 3);(1, 4);(2, 3);(2, 4);(3, 4)

NB. Vertex Values
NB. Connect Parameters

require 'gl3'

POPNGL=: 0 : 0
pc popngl;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 0 0 200 200;cc g isigraph ws_clipchildren ws_clipsiblings rightmove
bottommove;
xywh 203 30 34 11;cc RunN button;
xywh 202 7 34 11;cc RunT button;
xywh 202 93 34 11;cc Clear button;
pas 0 0;
rem form end;
)

create=: popngl_run

popngl_run =: 3 : 0
wd :: ] 'psel a;pclose'
wd POPNGL
formhwnd=: wd'qhwndp'
NB. form here opengl -----
glARC ''
R =: 0 0 0

```



```

SC =: 1
LS =: 0
Hid =: 0
RL =: 0
DB =: 0
Dbase =: dl_base_ NB. from base script program
glafont 'arial 30'
glusefontbitmaps 0 32 26 32
wd 'pshow;ptop'
wd 'pshow;'
)

destroy=: 3 : 0
wd'pclose'
codestroy''
)

popngl_cancel=:popngl_cancel_button=:popngl_close=:destroy

formselect=: 3 : 'wd''psel ''',formhwnd'

adjpos =: 3 : 0
:
SIZ =. x
POS =. y.
)

NB. display the model picture =====
popngl_g_paint =: verb define
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT
glLineWidth 4.0
draw ''
drawtext''
glSwapBuffers ''
)

connect =: 3 : 0
:
PP =. x.
C =. y.
i =. 0
while. i < #C
do.
    glVertex (i{C) { PP

```

```

        i =. i + 1
end.
)

PBox =: ] ;._2 (0 : 0)
    1 1 1
   _1 1 1
   _1 _1 1
    1 _1 1
    1 1 _1
   _1 1 _1
   _1 _1 _1
    1 _1 _1
)

PBox =: ". PBox

qposs =: 3 : 0
:
    SIZ =. x.
    POZ =. y.
    i =. 1
    while. i <: 4
        do.
            glBegin GL_LINE_STRIP
            glVertex POZ + SIZ * 0{PD
            glVertex POZ + SIZ * i{PD
            glEnd ''
            i =. i + 1
        end.
    )

NB. from tree4.ijs =====
qfd =: 3 : '4 #. ^:(_1) y.'
dfq =: 3 : '4 #. y.'

quart =: 3 : 0
qfd i. 4^y.
)

qtopos =: 3 : 0
k =. y.
if. k = 0 do. 0, 0, 0 return. end.

```

```

QK =. quart k
QT =. QK { <"(1) PQ
if. k = 1 do. return. end.
QA =. 0.5 ^ i. k
i =. 0
Q =. ''
while. i < {. $ QK
  do.
    j =. 0
    QT_i =. > i{QT
    while. j < k
      do.
        Q =. Q, < (j{QA) * (j{QT_i)
        j =. j + 1
      end.
    i =. i + 1
  end.
Q =. ((#Q)%k), k) $ Q
,. <"(1) (+/"(2) @: >) Q
)

```

```
pick =: 4 : ' (4*x.) }. (4*x.+1) {. y.'
```

```
NB. from tree4.ijs =====
```

```
NB. Draw Vertex
```

```

draw =: verb define
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glTranslate 0, 0, 0
glRotate R ,. 3 3 $ 1 0 0 0
glScale 3#SC
if. LS = 0
  do.
    glPolygonMode GL_FRONT, GL_LINE NB. Paint line
  else.
    glPolygonMode GL_FRONT_AND_BACK, GL_FILL NB. Paint full
  end.
glPolygonMode GL_BACK, Hid{GL_LINE, GL_POINT NB. Hidden

```

```
NB. Plot Curve =====
```

```

glBegin GL_LINE_STRIP
NB. glLineWidth 200.0 should be defined outside glBegin - glEnd
COL =: 0 0 0 0
COL =: COL ; 0.75 0.5 0 0

```

```

COL =: COL , < 0    1  0  0
COL =: COL , < 1    0.7 0  0
COL =: COL , < 1    0   0  0
COL =: COL , < 1    0   1  0
j =. 0
while. j <:DB
  do.
    glColor > j { COL
      (0.5^j) qposs L:0 qtopos j
    j =. j + 1
  end.
)

```

```

NB. project the picture on the screen =====
popngl_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. Key-in Command
NB. key-in x, y, z, X, Y, Z for rotation =====
popngl_g_char =: verb define
k =. 0 { sysdata
R =: 360 | R + 5 * 'xyz' = 0 { sysdata
R =: 360 | R - 5 * 'XYZ' = 0 { sysdata
SC =: SC * 1 + 0.25 * 'l' = 0 { sysdata NB. larger
SC =: SC * 1 - 0.25 * 's' = 0 { sysdata NB. smaller
LS =: ('s' = k) { LS, -. LS
Hid =: ('h' = k) { Hid, -. Hid
glpaintx''
)

```

```

NB. indicate rotated angle values x, y, z in degree =====
drawtext =: verb define
glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glColor 0 0 0 0
glRasterPos _1 _2 0
glCallLists 5 " : R
)

```

```
popngl_help_button =: verb define
wd 'mb OpenGL *Press keys, x/X, y/Y, z/Z rotate, s: line or solid, h: line
hidden toggle.'
wd 'setfocus g'
)
```

```
popngl_RunN_button=: 3 : 0
DB =: DB + 1
wd 'setfocus g'
glpaintx ''
)
```

```
popngl_RunT_button=: 3 : 0
DB =: DB + 1
wd 'setfocus g'
glpaintx ''
)
```

```
popngl_Clear_button=: 3 : 0
glClearColor 0 0 0 1
glColor 1 0 0 1
glpaintx ''
)
```