

J の gl3-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—

西川 利男

先の「J の gl3-OpenGL によるグラフィックス—その 1」では OpenGL の基本の考え方とその J プログラミングについて解説した。続いて「その 2」として、さまざまに回転する正 8 面体と正 12 面体に適用してみた。

正多面体はその元をたどれば、古くユークリッドの幾何学原本の最後に登場し[1]、プラトンの正多面体と言われる。すなわち、正 4 面体 {3,3}、正 6 面体(立方体) {4,3}、正 8 面体 {3,4}、正 12 面体 {5,3}、正 20 面体 {3,5} の 5 種類がある。ここで、記号 {m,n} は Schläfli の記法と呼ばれ、n 個の正 m 角形が各 1 つの頂点に集まることを示し、これにより、正多面体は 5 種類しか存在しないことが証明される。

1. 正 8 面体 (OpenGL3. ijs)

正 8 面体では頂点の 6 つの (X, Y, Z) 座標値は簡単に求められる。

```
V0 =: 1 0 1
V1 =: 1 0 _1
V2 =: _1 0 _1
V3 =: _1 0 1
V4 =: 0 1 0
V5 =: 0 _1 0
```

次にグラフィックスとして表示する際、Solid と Wired の 2 つの方法があり、目的に応じて使いわけ。

これらは `glPolygonMode` のパラメータとして、つぎに指定して選ぶ。

Solid:

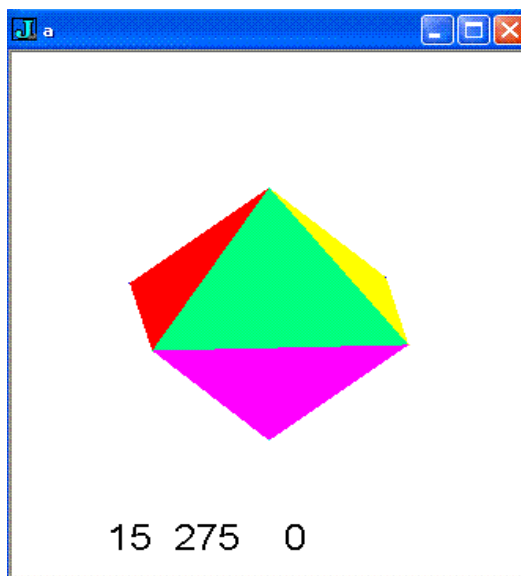
```
glPolygonMode GL_FRONT_AND_BACK, GL_FILL
```

Wired:

```
glPolygonMode GL_FRONT_AND_BACK, GL_LINE
```

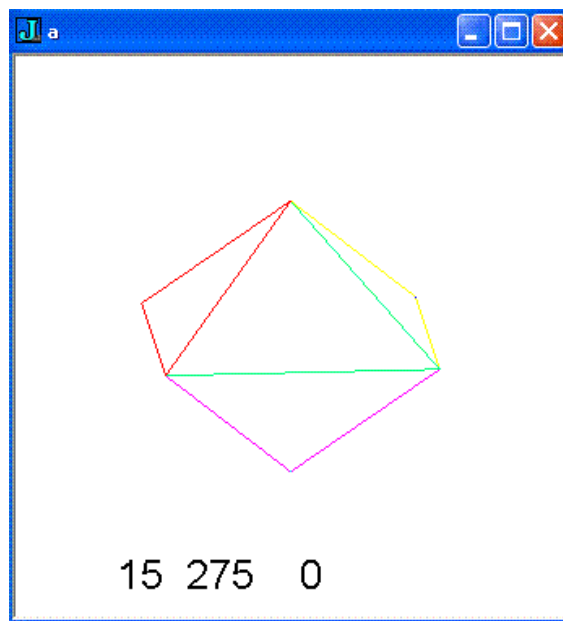
また、3次元グラフィックスでは見えない部分を隠す陰線処理が重要である。陰線処理を行うには、まず次のように指定する。

```
glPolygonMode GL_BACK, GL_POINT
```



多面体の各面を作るには、それぞれの頂点を結ぶことが必要である。ここで、陰線処理のためには、その順序の記述がポイントになる。座標軸に沿って、手前に見えるときに、角頂点が右ネジの向きになるよう配置する。これは次のようにコーディングされる。色の指定とともに、奥の頂点から配置した。

```
glBegin GL_TRIANGLES
  glColor 0.5 0.5 0.5 0
  glVertex V5
  glVertex V3
  glVertex V2
  glColor 1 0.5 0 0
  glVertex V5
  glVertex V2
  glVertex V1
  glColor 1 0 1 0
  glVertex V5
  glVertex V1
  glVertex V0
  glColor 0 1 1 0
  glVertex V5
  glVertex V0
  glVertex V3
  glColor 1 1 0 0
  glVertex V4
  glVertex V1
  glVertex V2
  glColor 0 0 1 0
  glVertex V4
  glVertex V2
  glVertex V3
  glColor 0 1 0.5 0
  glVertex V4
  glVertex V0
  glVertex V1
  glColor 1 0 0 0
  glVertex V4
  glVertex V3
  glVertex V0
glEnd ''
```



全体のプログラムには、そのほかいろいろなJのコーディングが必要だが、前報「その1」の説明から明らかであろう。
SolidとWiredのそれぞれの例を示した。

2. 正12面体(OpGLN5. ijs)

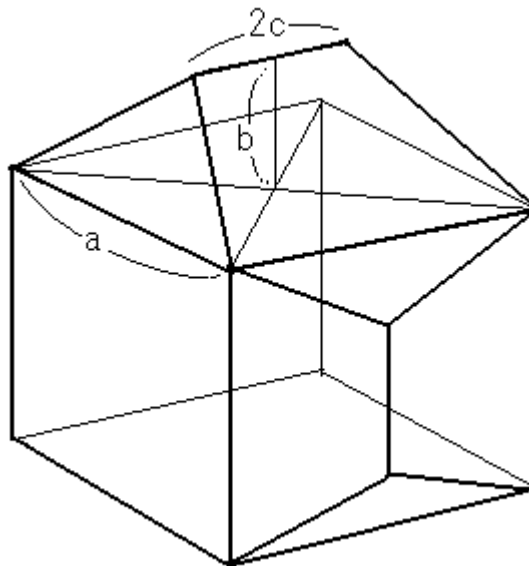
正12面体では頂点の位置座標を求めることが大変である。あちこち調べた末に、高木貞治先生と一松信先生の本に見つけることが出来た。

[1] 高木貞治「数学小景」p. 82-84, 岩波現代文庫(2002).

[2] 一松信「正多面体を解く」p. 16, 東海大学選書(1997).

[3] コセクター、銀林浩訳「幾何学入門」p. 38-40, p. 163-165, 明治図書(1965).

始めに一辺 a の立方体をとる。次にこの立方体の各面をなす正方形の中心から、面に垂直に外側に b だけ離れた点を取り、その点を通して適当な辺に平行な長さ $2b$ の線分の両端に点をとる。その点と元の立方体の面をなす正方形と結ぶ。つまり立方体の各面の外側に屋根をつける。その屋根を隣の屋根とつなぐ。すると2つの屋根は同一平面上になり、全体は正12面体になる。



このとき、 b と c は a により次の式で計算される。

$$b = \frac{1}{4}(\sqrt{5} - 1)a, \quad c = \frac{1}{8}(\sqrt{5} - 1)a$$

したがって、20個の頂点座標は以下ようになる。

NB. Dodecahedron Vertex Data

C =: ((%:5) - 1)%4

B =: 1 + ((%:5) - 1)%4

Vp =: 1 1 1

Vp =: Vp, 1 1 _1

Vp =: Vp, _1 1 _1

Vp =: Vp, _1 1 1

Vp =: Vp, 1 _1 1

Vp =: Vp, 1 _1 _1

Vp =: Vp, _1 _1 _1

```

Vp =: Vp,      _1  _1  1
Vp =: Vp,      0,   B,  C
Vp =: Vp,      0,   B, -C
Vp =: Vp,      (-C), 0,  B
Vp =: Vp,      C,   0,  B
Vp =: Vp,      B,   C,  0
Vp =: Vp,      B, (-C), 0
Vp =: Vp,      C,   0, -B
Vp =: Vp,      (-C), 0, -B
Vp =: Vp,      (-B), C,  0
Vp =: Vp,      (-B), (-C), 0
Vp =: Vp,      0, (-B),  C
Vp =: Vp,      0, (-B), (-C)
Vp =: 20 3$Vp

```

つづいて色を含めた頂点の結合は次の定義で決められる。

```

drawbox=:verb define
RED      polygon 0 8 3 10 11 {Vp NB. あか
GREEN    polygon 0 11 4 13 12 {Vp NB. みどり
BLUE     polygon 3 16 17 7 10 {Vp NB. あお
COLX     polygon 4 11 10 7 18 {Vp NB. き
COLY     polygon 1 9 8 0 12 {Vp NB. みずいろ
COLZ     polygon 1 12 13 5 14 {Vp NB. むらさき
COLP     polygon 5 19 18 4 13 {Vp NB. ちゃいろ
COLQ     polygon 1 14 15 2 9 {Vp NB. きみどり
COLR     polygon 2 15 6 17 16 {Vp NB. あかむらさき
COLS     polygon 6 15 14 5 19 {Vp NB. こいあお
COLT     polygon 3 8 9 2 16 {Vp NB. そらいろ
COLU     polygon 6 19 18 7 17 {Vp NB. はいいろ
)

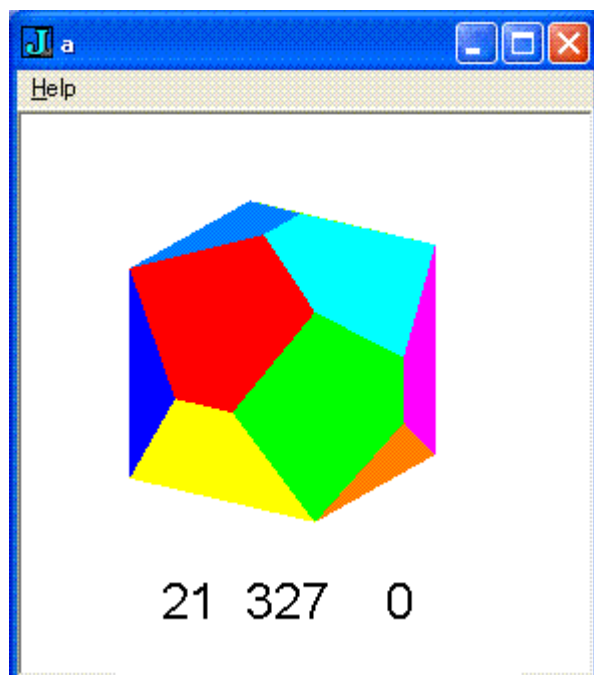
```

ここで関数 polygon を呼んでいる。

```

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

```



正12面体(OpGLN5.ijs) プログラム・リスト

NB. OpGLN5.ijs / Dodecahedron(Pentagonal 12-face Solid)

NB. 2009/9/1 by T. Nishikawa

NB. 2009/9/14 OK!

```
require 'gl3'

A=: noun define
pc a closeok;
menupop "&Help";
menu help "&Help" "" "" "";
menupopz;
xywh 0 0 200 200;cc g isigraph ws_clipchildren ws_clipsiblings rightmove
bottommove;
pas 0 0;
rem form end;
)

run =: a_run
a_run=: verb define
wd A
glaRC'' NB. enable gl3-OpenGL
R =: 0 0 0
glaFont 'arial 30'
glaUseFontBitmaps 0 32 26 32
wd 'pshow;ptop'
)

a_g_char =: verb define
R =: 360 | R + 3 * 'xyz' = 0 { sysdata NB. key_in 'x', rotate around x-axis
R =: 360 | R - 3 * 'XYZ' = 0 { sysdata NB. key_in 'X', rotate around x-axis g
glpaintx''
)

a_g_size=:verb define
wh=. glqwh''
glViewport 0 0,wh
glMatrixMode GL_PROJECTION
glLoadIdentity''
NB. gluPerspective 30, (%/wh),5 15 NB. for Perspective Projection
glOrtho _3 3 _3 3 _3 3 NB. for Ortho Projection
)

a_g_paint =: verb define
glClearColor 1 1 1 0 NB. back ground white
```

```

glClear GL_COLOR_BUFFER_BIT + GL_DEPTH_BUFFER_BIT
glEnable GL_DEPTH_TEST
glMatrixMode GL_MODELVIEW
glLoadIdentity''
glTranslate 0 0 0          NB. for Ortho Projection
NB. glTranslate 0 0 _10    NB. for Perspective Projection
glRotate R ,. 3 3 $ 1 0 0 0
NB. glPolygonMode GL_FRONT_AND_BACK, GL_LINE
glPolygonMode GL_FRONT_AND_BACK, GL_FILL
drawbox ''
drawtext ''
glSwapBuffers ''
)

```

NB. Dodecahedron Vertex Data

```

C =: ((%:5) - 1)%4
B =: 1 + ((%:5) - 1)%4
Vp =:      1      1      1
Vp =: Vp,    1      1     _1
Vp =: Vp,   _1      1     _1
Vp =: Vp,   _1      1      1
Vp =: Vp,    1     _1      1
Vp =: Vp,    1     _1     _1
Vp =: Vp,   _1     _1     _1
Vp =: Vp,   _1     _1      1
Vp =: Vp,    0,     B,     C
Vp =: Vp,    0,     B,    -C
Vp =: Vp,   (-C),  0,     B
Vp =: Vp,    C,     0,     B
Vp =: Vp,    B,     C,     0
Vp =: Vp,    B,   (-C),    0
Vp =: Vp,    C,     0,    -B
Vp =: Vp,   (-C),  0,    -B
Vp =: Vp,   (-B),  C,     0
Vp =: Vp,   (-B), (-C),    0
Vp =: Vp,    0,   (-B),    C
Vp =: Vp,    0,   (-B), (-C)
Vp =: 20 3$Vp

```

NB. Color Data

```

RED=: 1 0 0
GREEN=: 0 1 0
BLUE=: 0 0 1
COLX=: 1 1 0
COLY=: 0 1 1

```

```

COLZ=: 1 0 1
COLP=: 1 0.5 0
COLQ=: 0.5 1 0
COLR=: 1 0 0.5
COLS=: 0.5 0 1
COLT=: 0 0.5 1
COLU=: 0.5 0.5 0.5

```

NB. OK version

```
drawbox=: verb define
```

```

RED      polygon 0 8 3 10 11 {Vp NB. あか
GREEN    polygon 0 11 4 13 12 {Vp NB. みどり
BLUE     polygon 3 16 17 7 10 {Vp NB. あお
COLX     polygon 4 11 10 7 18 {Vp NB. き

COLY     polygon 1 9 8 0 12 {Vp NB. みずいろ /頂点でない点から結ぶ
COLZ     polygon 1 12 13 5 14 {Vp NB. むらさき
COLP     polygon 5 19 18 4 13 {Vp NB. ちゃいろ /頂点でない点から結ぶ

COLQ     polygon 1 14 15 2 9 {Vp NB. きみどり
COLR     polygon 2 15 6 17 16 {Vp NB. あかむらさき
COLS     polygon 6 15 14 5 19 {Vp NB. こいあお
COLT     polygon 3 8 9 2 16 {Vp NB. そらいろ
COLU     polygon 6 19 18 7 17 {Vp NB. はいいろ /頂点でない点から結ぶ
)

```

```
drawtext =: verb define
```

```

glMatrixMode GL_MODELVIEW
glLoadIdentity ''
glColor 0 0 0 0
glRasterPos _1.2 _2.9 0
glCallLists 5 " : R NB. indicate X, Y, Z rotated angles in deg
)

```

```
a_help_button=: verb define
```

```

wd' mb OpenGL *Press x y z to rotate.'
wd 'setfocus g'
)

```