

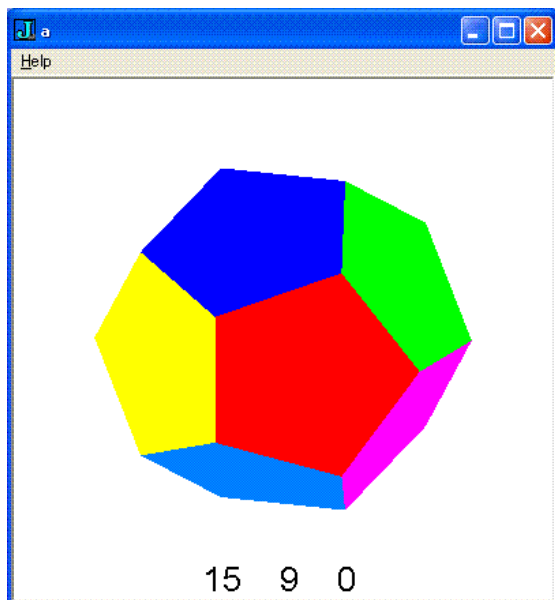
JのOpenGLグラフィックスーその5 ー正12面体と正20面体を動かすー

西川 利男

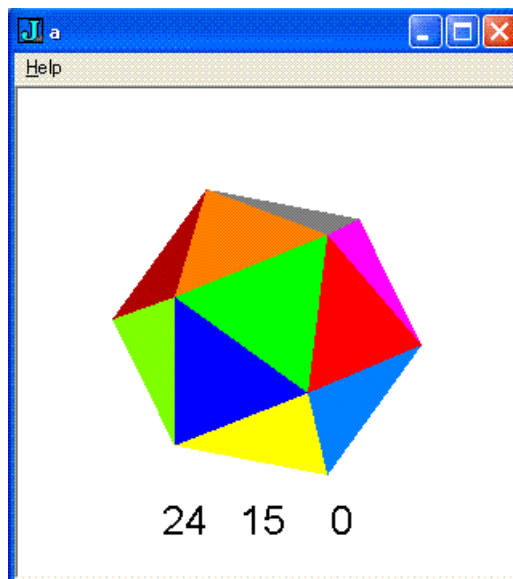
正12面体と正20面体との頂点座標が別報[1]のように計算されたので、それを用いてJのOpenGLにより3Dグラフィックス図形を描き、いろいろ動かしてみる。

1. 正12面体と正20面体のJプログラム (J402版, J602版) と実行例

同じJのOpenGLプログラムでも、J402版の方がOpenGLの基本の操作に直接対応していて筆者の好みに合っている。一方、J602版ではクラスファイルを利用するオブジェクト指向となっている。両方のプログラムを作成した。具体的な実際のプログラムコードは最後に示す。



0
pen
GL
の
操
作
部
分



は前回報告したもの[2]と同じである。座標の計算方法は、正12面体については前回と異なり今回の改良版[1]を用い、さらに正20面体について新たに加えたものである。とくに留意すべき点は、正多面体を構成する正多角形の座標の結合順序の指定をどうするかであるが、次にのべる。

[1] 西川利男「JのOpenGLグラフィックスーその4、正12面体と正20面体との頂点座標の計算」JAPLA シンポジウム資料 2009/12/5

[2] 西川利男「Jのgl3-OpenGLグラフィックスーその2、正8面体と正12面体を動かす」JAPLA 研究会資料 2009/9/29

2. 正12面体の頂点座標の順序付け

まず始めに、一辺2の正12面体の計算された頂点座標の値はつぎのとおりになる。

```

dodec 2
1. 7013      0  2. 22703
0. 525731  1. 61803  2. 22703
_1. 37638    1  2. 22703
_1. 37638    _1  2. 22703
0. 525731 _1. 61803  2. 22703
2. 75276     0  0. 525731
0. 850651  2. 61803  0. 525731
_2. 22703  1. 61803  0. 525731
_2. 22703 _1. 61803  0. 525731
0. 850651 _2. 61803  0. 525731
2. 22703 _1. 61803 _0. 525731
2. 22703  1. 61803 _0. 525731
_0. 850651  2. 61803 _0. 525731
_2. 75276     0 _0. 525731
_0. 850651 _2. 61803 _0. 525731
1. 37638     _1  _2. 22703
1. 37638     1  _2. 22703
_0. 525731  1. 61803  _2. 22703
_1. 7013     0  _2. 22703
_0. 525731 _1. 61803  _2. 22703

```

上の 20 個の座標の点を Z 軸の方向から見た図を示すと以下のようにになる。
そこで正 12 面体を構成する正五角形の各点を結びつける。

最初の結合は簡単である。

```
0 1 2 3 4
```

しかし、2 層目からは必ずしも簡単ではない。図のようにして

```
0 5 11 6 1
```

と行なわれる。続いて

```
1 6 12 7 2
```

```
2 7 13 8 3
```

```
3 8 14 9 4
```

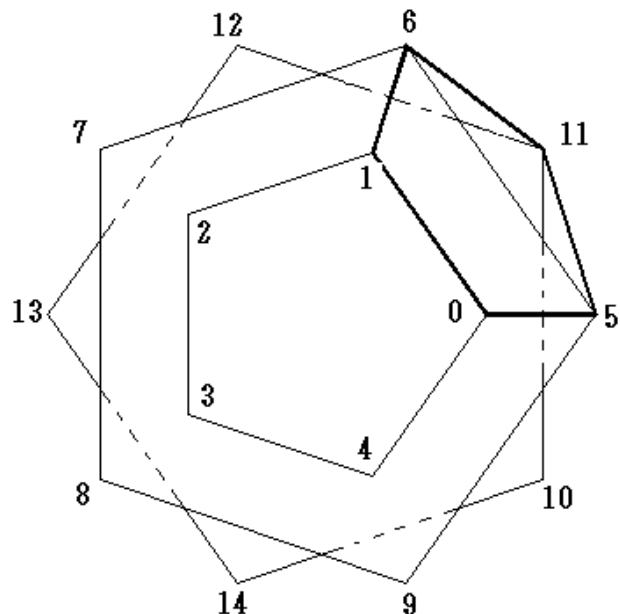
```
4 9 10 5 0
```

のようにになる。

このように座標の順序付けをどうするかが、ポイントになる。

同様にして、3 層目の点も結合することができる。

順序調整の処理の実際はプログラム dodec を見られたい[1]。



4. 正12面体と正20面体のJプログラム (J402版)

NB. Dodecahedron and Icosahedron

NB. OpenGL_Polyh. ijs

NB. 1st v 2009/9/1 by T. Nishikawa

NB. rev. 2009/9/14

NB. using revised polyhedron program 2009/11/7

NB. run 0 / run '' => Dodecahedron

NB. run 1 => Icosahedron

```
require 'gl3'
```

```
A=: noun define
```

```
pc a closeok;
```

```
menupop "&Help";
```

```
menu help "&Help" "" "" "";
```

```
menupopz;
```

```
xywh 0 0 220 200;cc g isigraph ws_clipchildren ws_clipsiblings rightmove
```

```
bottommove;
```

```
pas 0 0;
```

```
rem form end;
```

```
)
```

```
run =: a_run
```

```
a_run=: verb define
```

```
N =: y.
```

```
if. N = 0
```

```
do. Vd =: dodec 1.5
```

```
else. Vc =: icosah 2
```

```
end.
```

```
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 NB. wire frame
glPolygonMode GL_FRONT_AND_BACK, GL_FILL
if. N = 0
do. draw_dodec ''
else. draw_icosah ''
end.
drawtext ''
glSwapBuffers ''
)

```

NB. Color Data

```

COLA=: 1 0 0
COLB=: 0 1 0
COLC=: 0 0 1
COLD=: 1 1 0
COLE=: 0 0.5 1
COLF=: 1 0 1
COLG=: 1 0.5 0
COLH=: 0.5 1 0
COLI=: 1 0 0.5
COLJ=: 0.5 0 1
COLK=: 0 0.5 1
COLL=: 0.5 0.5 0.5
COLM=: 0.7 0 0
COLN=: 0.7 1 0
COLO=: 0.7 0 1
COLP=: 0.7 1 1

```

```

COLQ=: 0.3 0.3 0
COLR=: 0.3 0.3 1
COLS=: 0.3 1 0.3
COLT=: 0.3 1 1

```

```

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'
)

```

```

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

```

NB. Dodecahedron Vertex

```
draw_dodec=:verb define
```

```

COLA      polygon  0  1  2  3  4  {Vd NB.  あか
COLB      polygon  0  5 11  6  1  {Vd NB.  みどり
COLC      polygon  1  6 12  7  2  {Vd NB.  あお
COLD      polygon  2  7 13  8  3  {Vd NB.  き
COLE      polygon  3  8 14  9  4  {Vd NB.  みずいろ
COLF      polygon  4  9 10  5  0  {Vd NB.  むらさき

COLG      polygon 19 14  9 10 15  {Vd NB.  ちゃいろ
COLH      polygon 18 13  8 14 19  {Vd NB.  きみどり
COLI      polygon 17 12  7 13 18  {Vd NB.  あかむらさき
COLJ      polygon 16 11  6 12 17  {Vd NB.  こいあお
COLK      polygon 15 10  5 11 16  {Vd NB.  そらいろ

COLL      polygon 15 19 18 17 16  {Vd NB.  はいいろ
)

```

NB. Icosahedron Vertex

```

draw_icosahedron=:verb define
COLA      polygon 0 1 2 {Vc NB. あか
COLB      polygon 0 2 3 {Vc NB. みどり
COLC      polygon 0 3 4 {Vc NB. あお
COLD      polygon 0 4 5 {Vc NB. き
COLE      polygon 0 5 1 {Vc NB. みずいろ

COLF      polygon 1 2 7 {Vc NB. むらさき
COLG      polygon 2 3 8 {Vc NB. ちゃいろ
COLH      polygon 3 4 9 {Vc NB. きみどり
COLI      polygon 4 5 10 {Vc NB. あかむらさき
COLJ      polygon 5 1 6 {Vc NB. こいあお

COLK      polygon 1 6 7 {Vc NB. そらいろ
COLL      polygon 2 7 8 {Vc NB. はいいろ
COLM      polygon 3 8 9 {Vc NB.
COLN      polygon 4 9 10 {Vc NB.
COLO      polygon 5 10 6 {Vc NB.

COLP      polygon 6 7 11 {Vc NB.
COLQ      polygon 7 8 11 {Vc NB.
COLR      polygon 8 9 11 {Vc NB.
COLS      polygon 9 10 11 {Vc NB.
COLT      polygon 10 6 11 {Vc NB.
)

```

正多面体の頂点座標の計算プログラム、polyh, dodec, icosahedronなどは省略した。

以下の別報を見られたい。

[1] 西川利男「JのOpenGLグラフィックスーその4、正12面体と正20面体との頂点座標の計算」JAPLA シンポジウム資料 2009/12/5

5. 正12面体と正20面体のJプログラム (J602版)

NB. OpenGL_polyh.ijs

NB. run 0 / run '' => Dodecahedron

NB. run 1 => Icosahedrn

NB. rotate => key-in: x/X, y/Y, z/Z

```
require'opengl gl3'  
coinsert'jgl3'
```

```
A=: 0 : 0  
pc a closeok;  
xywh 0 0 220 200;cc g isigraph opengl rightmove bottommove;  
pas 0 0;  
rem form end;  
)
```

```
run=: a_run  
a_run=: 3 : 0  
N =: y  
if. N = 0  
  do. Vd =: dodec 0.4  
  else. Vc =: icosahedrn 0.5  
end.  
wd A  
ogl=: ''conew'jzopengl'  
R=: 0 0 0  
wd'pshow;  
)
```

```
a_close=: 3 : 0  
destroy__ogl''  
wd'pclose'  
)
```

```
a_g_paint=: 3 : 0  
RC =: rc__ogl ''  
if. RC do. g_draw_init wh__ogl end.  
g_draw''  
show__ogl''  
)
```

```
a_g_char =: 3 : 0  
wd'psel a'
```



```

R=: 360 | R + 2 * 'xyz' = {. sysdata
R=: 360 | R - 2 * 'XYZ' = {. sysdata
a_g_paint''
)

g_draw_init=: 3 : 0
glViewport 0 0, y
('arial';30) glaUseFontBitmaps__ogl 32 95 32
glMatrixMode GL_PROJECTION
glLoadIdentity''
glOrtho _1 1 _1 1 _1 1
NB. gluPerspective 30, (%/y), 1 10
)

g_draw=: 3 : 0
glClearColor 1 1 1 0
glClear GL_COLOR_BUFFER_BIT + GL_DEPTH_BUFFER_BIT
glEnable GL_DEPTH_TEST
glMatrixMode GL_MODELVIEW
glLoadIdentity''
glColor 0 0 0 0
glRasterPos _0.4 _0.7 0
glaCallLists ' X Y Z'
glRasterPos _0.4 _0.9 0
glaCallLists 5 ": R
glTranslated 0 0 0
glRotated R ,. 3 3 $ 1 0 0 0
NB. glPolygonMode GL_FRONT_AND_BACK, GL_LINE NB. Wired
if. N = 0
do. draw_dodec ''
else. draw_icosah ''
end.
)

polygon=: 4 : 0
glColor4d 4{x, 1
glBegin GL_POLYGON
glVertex y
glEnd ''
)

NB. Color Data
COLA=: 1 0 0
COLB=: 0 1 0

```

```

COLC=: 0 0 1
COLD=: 1 1 0
COLE=: 0 0.5 1
COLF=: 1 0 1
COLG=: 1 0.5 0
COLH=: 0.5 1 0
COLI=: 1 0 0.5
COLJ=: 0.5 0 1
COLK=: 0 0.5 1
COLL=: 0.5 0.5 0.5
COLM=: 0.7 0 0
COLN=: 0.7 1 0
COLO=: 0.7 0 1
COLP=: 0.7 1 1
COLQ=: 0.3 0.3 0
COLR=: 0.3 0.3 1
COLS=: 0.3 1 0.3
COLT=: 0.3 1 1

```

NB. Dodecahedron Vertex

draw_dodec=:verb define

```

COLA      polygon 0 1 2 3 4 {Vd NB. あか

COLB      polygon 0 5 11 6 1 {Vd NB. みどり
COLC      polygon 1 6 12 7 2 {Vd NB. あお
COLD      polygon 2 7 13 8 3 {Vd NB. き
COLE      polygon 3 8 14 9 4 {Vd NB. みずいろ
COLF      polygon 4 9 10 5 0 {Vd NB. むらさき

COLG      polygon 19 14 9 10 15 {Vd NB. ちゃいろ
COLH      polygon 18 13 8 14 19 {Vd NB. きみどり
COLI      polygon 17 12 7 13 18 {Vd NB. あかむらさき
COLJ      polygon 16 11 6 12 17 {Vd NB. こいあお
COLK      polygon 15 10 5 11 16 {Vd NB. そらいろ

COLL      polygon 15 19 18 17 16 {Vd NB. はいいろ
)

```

NB. Icosahedron Vertex

draw_icosah=:verb define

```

COLA      polygon 0 1 2 {Vc NB. あか
COLB      polygon 0 2 3 {Vc NB. みどり
COLC      polygon 0 3 4 {Vc NB. あお
COLD      polygon 0 4 5 {Vc NB. き
COLE      polygon 0 5 1 {Vc NB. みずいろ

```

COLF	polygon	1	2	7	{Vc	NB.	むらさき
COLG	polygon	2	3	8	{Vc	NB.	ちゃいろ
COLH	polygon	3	4	9	{Vc	NB.	きみどり
COLI	polygon	4	5	10	{Vc	NB.	あかむらさき
COLJ	polygon	5	1	6	{Vc	NB.	こいあお
COLK	polygon	1	6	7	{Vc	NB.	そらいろ
COLL	polygon	2	7	8	{Vc	NB.	はいいろ
COLM	polygon	3	8	9	{Vc	NB.	
COLN	polygon	4	9	10	{Vc	NB.	
COLO	polygon	5	10	6	{Vc	NB.	
COLP	polygon	6	7	11	{Vc	NB.	
COLQ	polygon	7	8	11	{Vc	NB.	
COLR	polygon	8	9	11	{Vc	NB.	
COLS	polygon	9	10	11	{Vc	NB.	
COLT	polygon	10	6	11	{Vc	NB.	

)

正多面体の頂点座標の計算プログラム、polyh, dodec, icosoなどは省略した。

以下の別報を見られたい。

[1] 西川利男「JのOpenGLグラフィックスーその4、正12面体と正20面体との頂点座標の計算」JAPLAシンポジウム資料 2009/12/5