Stonehenge
プログラムの説明

1. シーンや描画、プログラムに関する説明

・ 何を作ったか

ロンドンの世界遺産である「ストーンヘンジ」の崩れる前の本来の形と想像される形(図１)を作った.

図１. 参考画像

・ プログラムの概要

今回使用している図形はすべて，関数 Shape makePrism(int N)を用いて N=4 の直方体(４角柱)に設定している．このプログラムは①～④の 4 部に分けられる.

① 外側の円陣状にならんだ岩の柱部分を shape1 とし，内側の面がそれぞれ円の中心を向くように回転しながら，for 文を用いて円周上を等間隔に描画する. その上に横たわって乗っている岩を shape2 とし，shape1 と同様に内側の面がそれぞれ円の中心を向くように回転をしながら，for 文を用いてすべての柱と柱に橋を架けるように描画する.

② 内側の 5 つの門型に積まれた岩の柱部分の 2 本を shape3，その上に乗っている岩を shape4 とし，shape1 と shape2 と同様に円周上を任意の位置に for 文を用いて描画する.

③ 画像の中央にある一回り小さい岩を shape5 とし，任意の座標と向きに設定し描画する.

④ shape1~shape5 のメモリ開放

①，②について，shape1 から shape4 の座標を決定する際，sin 関数や cos 関数を使用し，滑らかな円形を表現した.

・ プログラムの説明(　2. プログラムと対応している　)

2~8 行目：視点の座標と向き，上向きベクトルを決定

① 10，11 行目：shape1 を stretch(&shape1, 0.2, 0.2, 1.0)と定める．
12 行目：側面が x 軸 y 軸方向に向くように z 軸を軸に 45° 回転
13，14 行目：shape2 を shape1 で作る柱にちょうど乗るように大きさを stretch(&shape2, 0.2, 0.2, 0.9)と定める．
15，16 行目：横たわるように z 軸を軸に 45° 回転した後 x 軸を軸に-90° 回転
18，19 行目：shape1，shape2 の for 文内で用いる回転角度のパラメータをそれぞれ k，l とする．
○20~34 行目は i を 0 から 2*M_PI まで M_PI/10 を足しながら繰り返す
21，28 行目：shape1，shape2 を描画する際の向きにパラメータを用いて z 軸回転
22，29 行目：shape1，shape2 を描画する際の座標に移動．(shape2 は shape1 の上に綺麗に乗るように z 座標を+1.0 + sqrt(0.02)している．)
23，30 行目：shape1，shape2 を描画
24，25，31，32 行目：21，22，28，29 行目と同じ数だけマイナスすることによって，shape1，shape2 の座標を原点(0, 0, 0)に戻し，元の角度に戻す．
26，33 行目：回転角度の変化のためにパラメータをそれぞれ適度に増加させる．(試行錯誤の結果，k++，l+=2 となった．)

② 36~38 行目：shape3 を stretch(&shape3, 0.2, 0.2, 1.0)と定め，側面が x 軸 y 軸方向に向くように z 軸を軸に 45° 回転させる．
39~42 行目：shape4 を shape3 で作る柱にちょうど乗るように大きさを stretch(&shape4, 0.2, 0.2, 0.5)と定め，横たわるように z 軸を軸に 45° 回転した後 x 軸を軸に-90° 回転させる．
44，45 行目：shape3，shape3 の for 文内で用いる回転角度のパラメータをそれぞれ m，n とする．
○46~64 行目は i を 0 から(14.0/9.0)*M_PI まで M_PI/9 を足しながら繰り返す
○47 行目：m%3!=0 のとき 47~61 行目のプログラムを実行する．
(shape3 を 2 個ずつ 1 個飛ばしに描画するために m%!=0 を条件とした)
48，49 行目：shape3 を描画する際の向きにパラメータを用いて z 軸回転させ，描画する際の座標に移動．
50 行目：shape3 を描画
51，52 行目：48，49 行目と同じ数だけマイナスすることによって，shape3

の座標を原点(0, 0, 0)に戻し，元の角度に戻す.

○54 行目：m%3==1 のとき 54~60 行目のプログラムを実行する.

(shape4 を 1 個ずつ 2 個飛ばしに描画するために m%!=0 を条件とした)

55，56 行目：shape4 を描画する際の向きにパラメータを用いて z 軸回転させ，描画する際の座標に移動.

50 行目：shape4 を描画

58, 59 行目:55, 56 行目と同じ数だけマイナスすることによって, shape4 の座標を原点(0, 0, 0)に戻し，元の角度に戻す.

62，63 行目：回転角度の変化のためにパラメータをそれぞれ適度に増加させる.（試行錯誤の結果，m++，n+=2 となった.）

③ 66，67 行目：shape5 を stretch(&shape5, 0.2, 0.2, 0.5)と定める.

68 行目：z 軸を軸に 15° 回転させる.

69 行目：shape5 を座標(-0.4, 0.3, 0)に移動させる.

70 行目：shape5 を描画

④ 71~75 行目：shape1 から shape5 分のメモリ解放

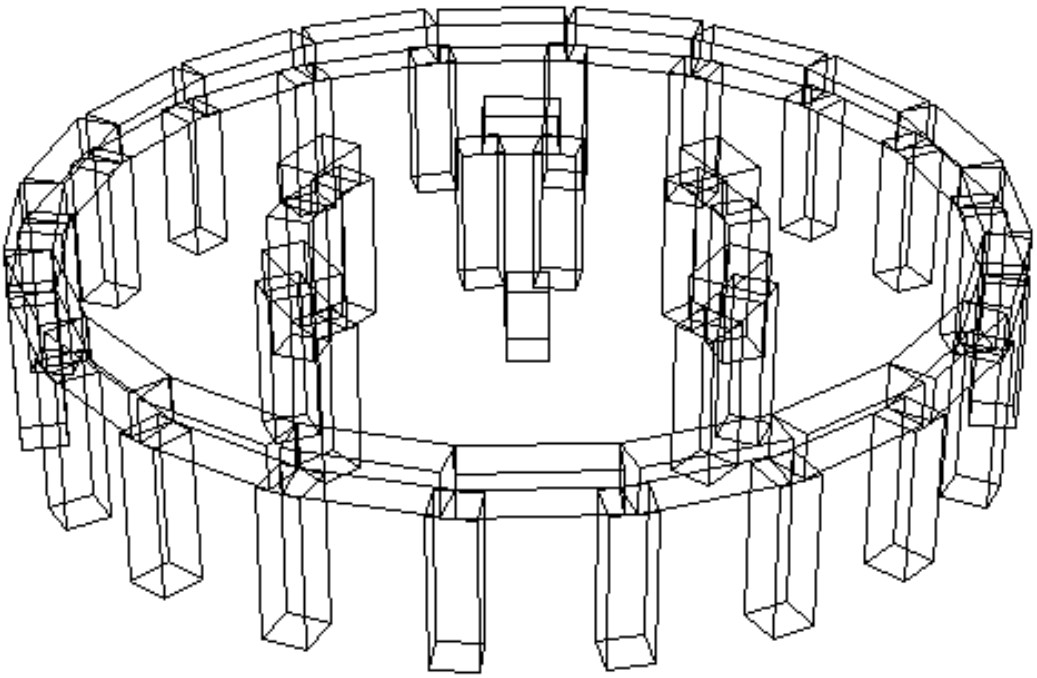
2.　プログラムリスト(OnDraw()関数)

```
1   void CComputerGraphicsReport02View::OnDraw(CDC* pDC){
2           Point3 V, D, U;
3           V.x = 0;   V.y = 0; V.z = 30;                  // 視点(0, 0, 30)
4           // V.x = 15;   V.y = -8.5; V.z = 10;           // 視点(15, -8.5, 10)
5           D.x = 0.0 - V.x; D.y = 0.0 - V.y; D.z = 0 - V.z;       // 見ている方角
6           U.x = 0; U.y = 1; U.z = 0;                     // 上向きベクトル
7           // U.x = 0; U.y = 0; U.z = 1;                  // 上向きベクトル
8           setTrans(V, D, U);
9           ///// ここから①
10          Shape shape1 = makePrism(4);   // shape1 を定義
11          stretch(&shape1, 0.2, 0.2, 1.0);
12          rotateZ(&shape1, 45);
13          Shape shape2 = makePrism(4);   // shape2 を定義
14          stretch(&shape2, 0.2, 0.2, 0.9);
15          rotateZ(&shape2, 45);
16          rotateX(&shape2, -90);
17
18          int k = 0;          // shape1 の回転角度のパラメータ
19          int l = 1;          // shape2 の回転角度のパラメータ
20          for (double i = 0 ; i < 2* M_PI ;   i+= M_PI/10) {
21                  rotateZ(&shape1, 18 * k);
22                  move(&shape1, 3*cos(i), 3*sin(i), 0);
23                  drawShape(pDC, &shape1);  // shape1 を描画
24                  move(&shape1, -3 * cos(i), -3 * sin(i), 0);
25                  rotateZ(&shape1, -18 * k);
26                  k++;
27
28                  rotateZ(&shape2, 9 * l);
29                  move(&shape2, 3 * cos(i), 3 * sin(i), 1.0 + sqrt(0.02));
30                  drawShape(pDC, &shape2);  // shape2 を描画
31                  move(&shape2, -3 * cos(i), -3 * sin(i), -1.0 - sqrt(0.02));
32                  rotateZ(&shape2, -9 * l);
33                  l += 2;
34          }
```

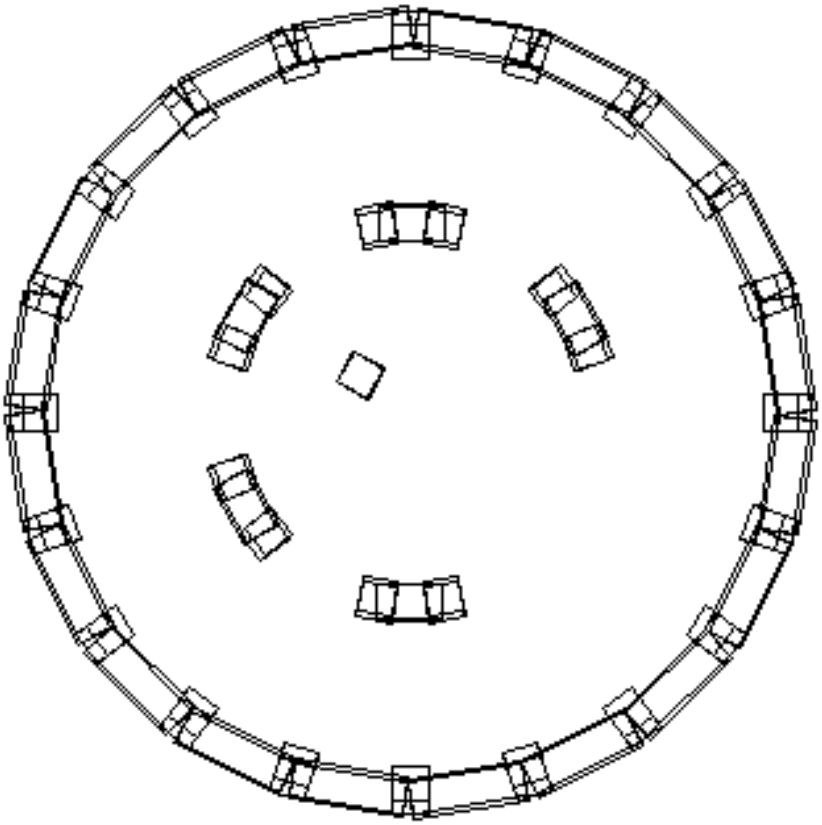
```
35          ///// ここから②
36          Shape shape3 = makePrism(4);    // shape3 を定義
37          stretch(&shape3, 0.2, 0.2, 1.0);
38          rotateZ(&shape3, 45);
39          Shape shape4 = makePrism(4);    // shape4 を定義
40          stretch(&shape4, 0.2, 0.2, 0.5);
41          rotateZ(&shape4, 45);
42          rotateX(&shape4, -90);
43
44          int m = 0;          // shape3 の回転角度のパラメータ
45          int n = 1;          // shape4 の回転角度のパラメータ
46          for (double i = 0; i < (14.0/9.0) * M_PI; i += M_PI / 9) {
47              if (m % 3 != 0) {
48                  rotateZ(&shape3, 20 * m);
49                  move(&shape3, 1.5 * cos(i), 1.5 * sin(i), 0);
50                  drawShape(pDC, &shape3);        // shape3 を描画
51                  move(&shape3, -1.5 * cos(i), -1.5 * sin(i), 0);
52                  rotateZ(&shape3, -20 * m);
53
54                  if (m % 3 == 1) {
55                      rotateZ(&shape4,    10*n);
56                      move(&shape4, 1.5*cos(i), 1.5* sin(i), 1.0 + sqrt(0.02));
57                      drawShape(pDC, &shape4);  // shape4 を描画
58                      move(&shape4,-1.5*cos(i),-1.5*sin(i), -1.0- sqrt(0.02));
59                      rotateZ(&shape4,    -10*n);
60                  }
61              }
62              m++;
63              n += 2;
64          }
65          ///// ここから③
66          Shape shape5 = makePrism(4);    // shape5 を定義
67          stretch(&shape5, 0.2, 0.2, 0.5);
68          rotateZ(&shape5, 15);
69          move(&shape5, -0.4, 0.3, 0);
70          drawShape(pDC, &shape5);                    // shape5 を描画
```

```
71          ///// ここから④
72          deleteShape(&shape1);
73          deleteShape(&shape2);
74          deleteShape(&shape3);
75          deleteShape(&shape4);
76          deleteShape(&shape5);
77  }
```

3.　画像のプリントアウト



視点(15, -8.5, 10)、上向きベクトル(0, 0, 1)の画像



視点(0 , 0, 30)、上向きベクトル(0, 1, 0)からの画像