

```
import processing.opengl.*;
import IGeo.p.*;
import IGeo.c.*;
import IGeo.g.*;
import IGeo.u.*;
import IGeo.c.*;
import IGeo.g.*;
import IGeo.u.*;

size(480, 360, IGL);
IG.open("newsphere1.3dm");

ISurface[] surfaces = IG.surfaces();
for (ISurface surf:surfaces) {
  int unum=1, vnum=1;
  double uinc=1.0/unum, vinc=1.0/vnum;
  for (int i=0; i<unum; i++) {
    for (int j=0; j<vnum; j++) {
      IVec pt1 = surf.pt(i*vinc, j*vinc);
      IVec pt2 = surf.pt(i+1*vinc, j*vinc);
      IVec pt3 = surf.pt(i*vinc, (j+1)*vinc);
      IVec pt4 = surf.pt(i+1*vinc, (j+1)*vinc);

      //folding in U
      double ratio1 = IRandom.get(0.5,0.7);
      double ratio1a = IRandom.get(0.0,2);
      double ratio2 = IRandom.get(0.8,1);
      double ratio2a = IRandom.get(0.0,4);

      // getting point on U edge
      IVec pt5 = surf.pt(i+ratio1*vinc, j*vinc);
      IVec pt6 = surf.pt(i+ratio2*vinc, j*vinc);
      IVec pt7 = surf.pt(i+ratio1a*vinc, (j+1)*vinc);
      IVec pt8 = surf.pt(i+ratio2a*vinc, (j+1)*vinc);

      IVec rotAxis = pt5.diff(pt1);
      IVec rotCenter = pt1;
      double rotAngle = IRandom.get(-0.2, -0.4);

      pt5.rot(rotCenter, rotAxis, rotAngle);
      pt7.rot(rotCenter, rotAxis, rotAngle);
      pt6.rot(rotCenter, rotAxis, rotAngle);
      pt8.rot(rotCenter, rotAxis, rotAngle);
      pt2.rot(rotCenter, rotAxis, rotAngle);
      pt3.rot(rotCenter, rotAxis, rotAngle);

      IVec rotAxis2 = pt7.diff(pt5);
      IVec rotCenter2 = pt5;
      double rotAngle2 = IRandom.get(-0.8, -0.6);

      pt6.rot(rotCenter2, rotAxis2, rotAngle2);
      pt8.rot(rotCenter2, rotAxis2, rotAngle2);
      pt2.rot(rotCenter2, rotAxis2, rotAngle2);
      pt3.rot(rotCenter2, rotAxis2, rotAngle2);

      IVec rotAxis3 = pt6.diff(pt8);
      IVec rotCenter3 = pt8;
      double rotAngle3 = IRandom.get(0.3,0.6);

      pt2.rot(rotCenter3, rotAxis3, rotAngle3);
      pt3.rot(rotCenter3, rotAxis3, rotAngle3);

      IVec[] cps = new IVec[2][4];
      cps[0][0] = pt2;
      cps[0][1] = pt6;
      cps[0][2] = pt5;
      cps[0][3] = pt1;
      cps[1][0] = pt3;
      cps[1][1] = pt8;
      cps[1][2] = pt7;
      cps[1][3] = pt4;

      int udeg4 = 1, vdeg4 = 1;
      new ISurface(cps, udeg4, vdeg4).chr(0.7, ratio2*IRandom.get(0.1,0.8),0.2);

      surf.del();
    }
  }

  // folding in V
  double ratio1 = IRandom.get(0.0,2);
  double ratio1a = IRandom.get(0.6,0.8);
  double ratio2 = IRandom.get(0.2,0.4);
  double ratio2a = IRandom.get(0.8,1);

  // getting point on V edge
  IVec pt5 = surf.pt(i*vinc, (j+ratio1)*vinc);
  IVec pt6 = surf.pt(i*vinc, (j+ratio2)*vinc);
  IVec pt7 = surf.pt(i+1*vinc, (j+ratio1a)*vinc);
  IVec pt8 = surf.pt(i+1*vinc, (j+ratio2a)*vinc);

  IVec rotAxis = pt4.diff(pt1);
  IVec rotCenter = pt1;
  double rotAngle = IRandom.get(-0.3, -0.4);

  pt5.rot(rotCenter, rotAxis, rotAngle);
  pt7.rot(rotCenter, rotAxis, rotAngle);
  pt6.rot(rotCenter, rotAxis, rotAngle);
  pt8.rot(rotCenter, rotAxis, rotAngle);
  pt2.rot(rotCenter, rotAxis, rotAngle);
  pt3.rot(rotCenter, rotAxis, rotAngle);

  IVec rotAxis2 = pt7.diff(pt5);
  IVec rotCenter2 = pt5;
  double rotAngle2 = IRandom.get(-0.8,0);

  pt6.rot(rotCenter2, rotAxis2, rotAngle2);
  pt8.rot(rotCenter2, rotAxis2, rotAngle2);
  pt2.rot(rotCenter2, rotAxis2, rotAngle2);
  pt3.rot(rotCenter2, rotAxis2, rotAngle2);

  IVec rotAxis3 = pt6.diff(pt8);
  IVec rotCenter3 = pt8;
  double rotAngle3 = IRandom.get(0.7,1.0);

  pt2.rot(rotCenter3, rotAxis3, rotAngle3);
  pt3.rot(rotCenter3, rotAxis3, rotAngle3);

  IVec[] cps = new IVec[2][4];
  cps[0][0] = pt1;
  cps[0][1] = pt5;
  cps[0][2] = pt6;
  cps[0][3] = pt4;
  cps[1][0] = pt2;
  cps[1][1] = pt7;
  cps[1][2] = pt8;
  cps[1][3] = pt3;

  int udeg4 = 1, vdeg4 = 1;
  new ISurface(cps, udeg4, vdeg4).chr(IRandom.get(0.1,1), IRandom.get(0.1,0.8),0);

  surf.del();
}

//IG.save("newsphere1.outcolour.3dm");
```

