

```

import igeo.p.*;
import processing.opengl.*;
size (600, 600, IGL.GL);
ISurface[] surfaces = IG.surfaces();

IImageMap map1 = new IImageMap("map1.jpg");
IImageMap map2 = new IImageMap("map2.jpg");

ISurface surfB = surfaces[0];
ISurface surfA = surfaces[1];

ILayer layer1 = IGLayer("exterior triangular panelization");
ILayer layer4 = IGLayer("spikes");

int unum = 30, vnum = 30;
double uinc = 1.0/unum, vinc = 1.0/vnum;
for (int i=0; i<unum; i++) {
  for (int j=0; j<vnum; j++) {
    if ( (i+j)%2==0) {
      IVec ptA11 = surfA.pt( i * uinc, j * vinc );
      IVec ptA21 = surfA.pt( (i + 1) * uinc, (j - 1) * vinc );
      new ISurface(ptA11, ptA21, ptA22).clr( map3.clr( i * uinc, j * vinc ), layer[1] );
      new ISurface(ptB11, ptB21, ptB22).clr( map6.clr( i * uinc, j * vinc ), layer[1] );

      double structureOffset = 1;
      double structureRadius = 0.1;
      IVec ptB11off = surfB.pt( i * uinc, j * vinc, structureOffset );
      IVec ptB21off = surfB.pt( (i + 1) * uinc, (j - 1) * vinc, structureOffset );
      new ICylinder(ptB11off, ptB21off, structureRadius).clr( map3.clr( i * uinc, j * vinc ), layer[6] );
      new ICylinder(ptB11off, ptB21off, structureRadius).clr( map3.clr( i * uinc, (j + 1) * vinc ), layer[6] );

      double structure2Radius = 0.05;
      new ICylinder(ptB21, ptB21off, structure2Radius).clr( map3.clr( i * uinc, j * vinc ), layer[7] );
      new ICylinder(ptB22, ptB22off, structure2Radius).clr( map3.clr( i * uinc, (j + 1) * vinc ), layer[7] );

      double size = 0.125;
      IGL.squarePipe[ new IVec[] { ptA11, ptA21, ptA22 }, 1, true, size ].clr( 0, 1 - i * uinc, 1 - 0.05 * i, 3 ), layer[ layer2 ];

      double radius = 0.05;
      if ( i > 10 && i < 20 ) {
        new ICylinder(ptA11, ptB11, radius).clr( map2.clr( i * uinc, j * vinc ), layer[ layer3 ];
      }

      IVec pt1 = surfA.pt( i * uinc, j * vinc );
      IVec pt2 = surfA.pt( (i + 1) * uinc, j * vinc );

      double val1 = map1.get( i * uinc, j * vinc );
      double depth = -8 * val1;
      double dratio = (depth + 3) / 10;

      IVec pt1d = surfA.pt( i * uinc, j * vinc, depth );
      IVec pt2d = surfA.pt( (i + 1) * uinc, j * vinc, depth );

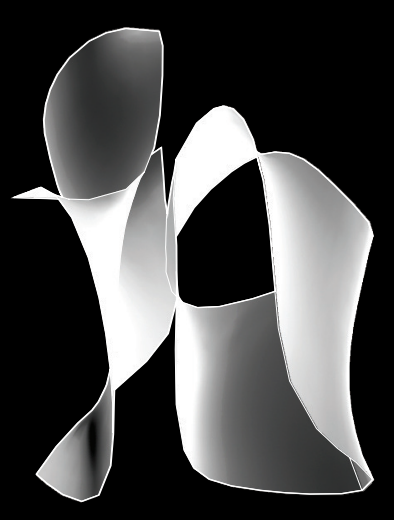
      IVec[] cpts4 = new IVec[3][2];
      cpts4[0][0] = pt4;
      cpts4[1][0] = pt4.mid(pt3);
      cpts4[2][0] = pt3;
      cpts4[0][1] = pt3m;
      cpts4[1][1] = pt3d;
      cpts4[2][1] = pt2m;
      new ISurface(cpts4, 2, 1).clr( map2.clr( i * uinc, j * vinc ), layer[5] );

      double val4 = map4.get( i * uinc, j * vinc );
      double spineDepth = -12 * val4 - 0.5;
      double spineWidth = 0.3;
      IVec spinePt1 = surfA.pt( (i + 0.6) * uinc, (j + 0.4) * vinc );
      new ICylinder(spinePt1, spinePt2, spineWidth, 0).clr( map3.clr( (i + 0.6) * uinc, (j + 0.4) * vinc ), layer[4] );

      double structureOffset2 = 3;
      double structureRadius2 = 0.1;
      IVec ptB11off2 = surfB.pt( i * uinc, (j - 1) * vinc, structureOffset2 );
      new ICylinder(ptB11off2, ptB21off2, structureRadius2).clr( map5.clr( i * uinc, j * vinc ), layer[6] );

      double structure2Radius2 = 0.1;
      new ICylinder(ptB21off, ptB21off2, structure2Radius2).clr( map5.clr( i * uinc, j * vinc ), layer[7] );
    }
  }
}
surfA.del();
surfB.del();

```



Input Geometry

Interstitial Space Connectors

Interior Diagrid Structure

Exterior Panelization

Exterior Panelization

Exterior Diagrid Structure

Facade Apertures

Spike Articulation

Exterior Diagrid Structure

color

color

color

depth

color

depth

color

