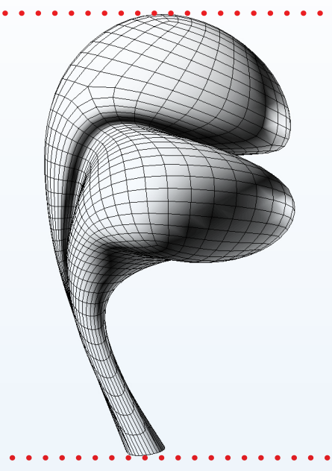
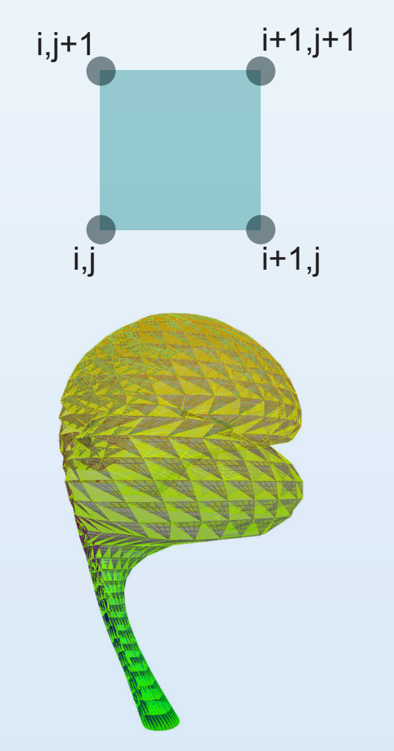
Satoru Sugihara - Coding Form - Final by: Pablo Osorio, 2012

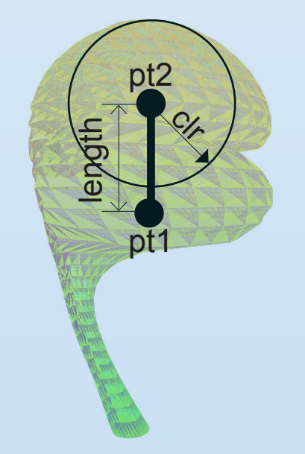
```
import processing.opengl.*;
import igeo.*;
void setup() {
  size(800, 800, IG.GL);
  IG.duration(200);
  IG.open("003.3dm"); ISurface surff = IG.surface(0);
  new LineAgent(new IVec(0, 0, 0), new IVec(1, 0, 0), surff);
  new MyBoundary();new LineAgent2(new IVec(932.77, 1954.85,
2210.91), new IVec(0, 0, 1));
```



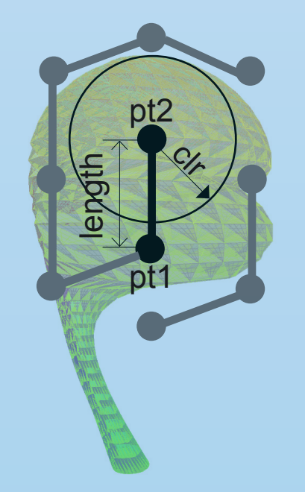
```
  ISurface[] surfs = IG.surfaces();
  for ( ISurface surf : surfs ) {

    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++) {
        IVec pt11 = surf.pt( i*uinc, j*vinc );
        IVec pt21 = surf.pt( (i+1)*uinc, j*vinc );
        IVec pt12 = surf.pt( i*uinc, (j+1)*vinc );
        IVec pt22 = surf.pt( (i+1)*uinc, (j+1)*vinc );
        new ISurface(pt11, pt21, pt12).clr(0, i*uinc, 5);
        new ISurface(pt12, pt21, pt22).clr(.4, i*uinc, j*vinc);
      }
    }
    surf.del();
  }
```



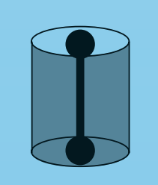
```
static class LineAgent extends IAgent {
  static double length = .01; //length in u-v space, less than 1.0
  static double clearance = 0.005; //less than length

  IVec pt1, pt2;
  boolean isColliding=false;
  ISurface surf;

  LineAgent(IVec pt, IVec dir, ISurface s) {
    pt1 = pt;
    pt2 = pt.dup().add(dir.dup().len(length));
    surf = s;
  }
```



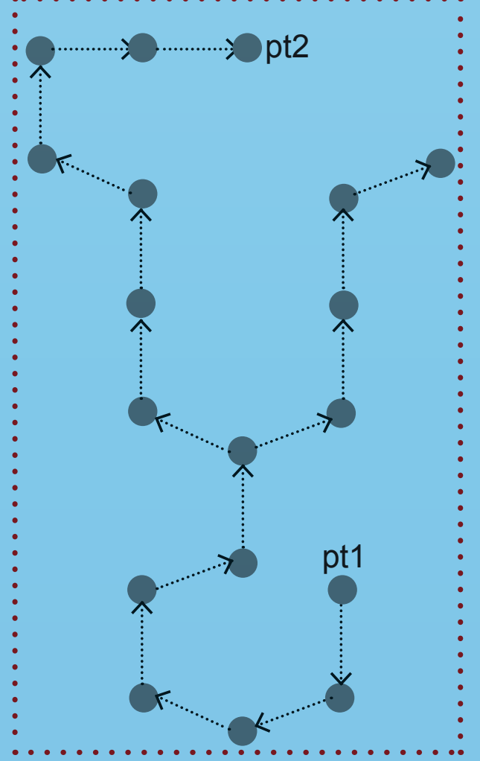
```
  void interact(ArrayList < IDynamics > agents){
    super.interact(agents);
    if(time == 0){ //only in the first time
      for(int i=0; i < agents.size() && !isColliding; i++){
        if(agents.get(i) instanceof LineAgent){
          LineAgent lineAgent =
            (LineAgent)agents.get(i);
          if(lineAgent != this){ //agents include "this"
            //checking clearance of end point
            if(lineAgent.pt2.dist(pt2) < clearance){
              isColliding=true;
            }
          }
        }
      }
    }
  }
  void update() {
    super.update();

    // is inside surface?
    if (pt2.x < 0.0||pt2.x > 1.0||pt2.y < 0.0||pt2.y > 1.0) {
      isColliding = true;
    }

    if (isColliding) {
      del();
    }
    else if (time == 0) { //if not colliding
      IVec surfPt1 = surf.pt(pt1.x, pt1.y);
      IVec surfPt2 = surf.pt(pt2.x, pt2.y);
      new ICurve(surfPt1, surfPt2).clr(0);
```

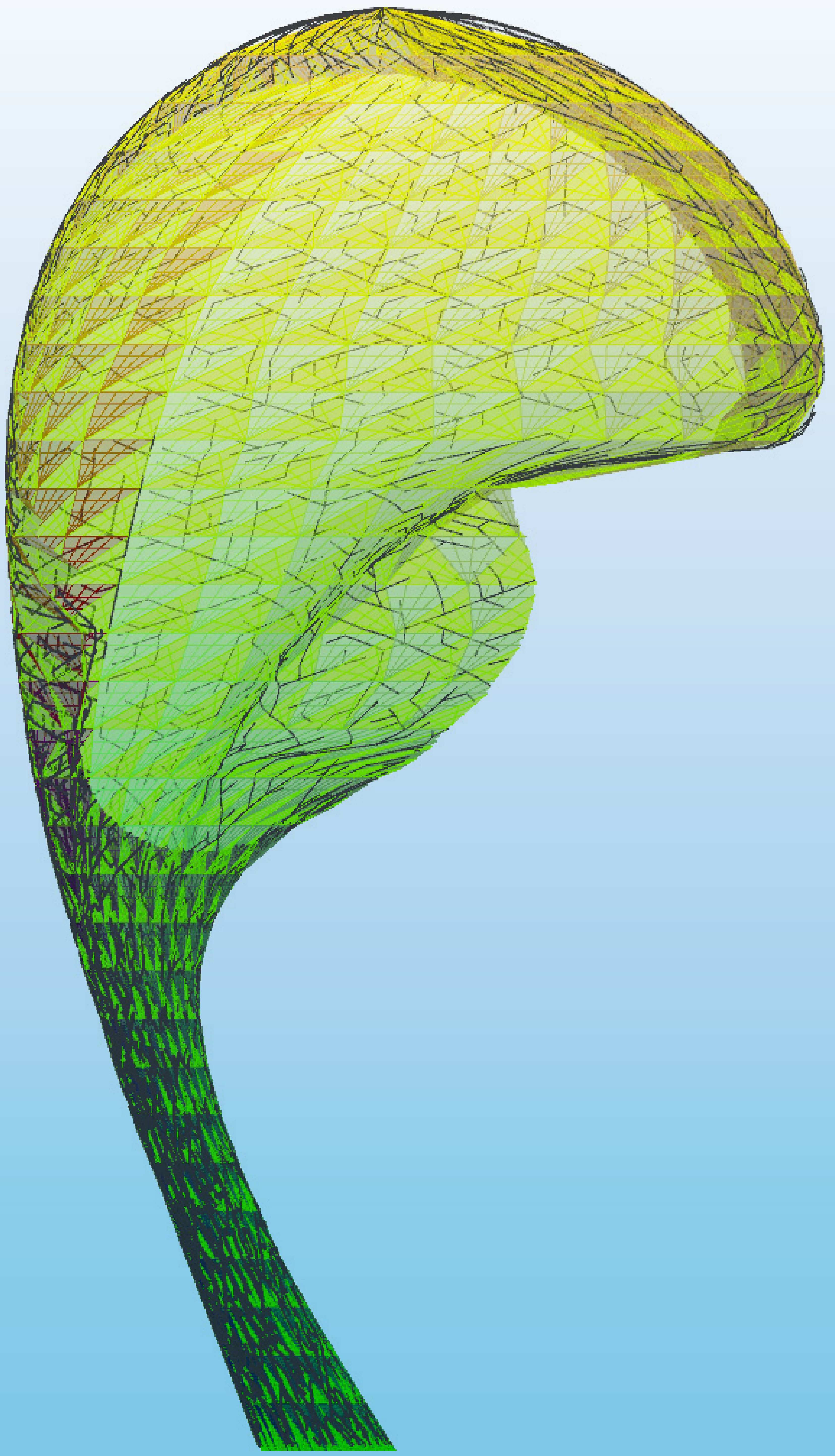


```
      double thickness = 10.2 - (double)IG.time()/IG.duration()*20.2;
      if (thickness< 0.2) {
        thickness = 0.2;
      }
      IG.squarePipe(surfPt1, surfPt2, thickness).clr(0.3);
```



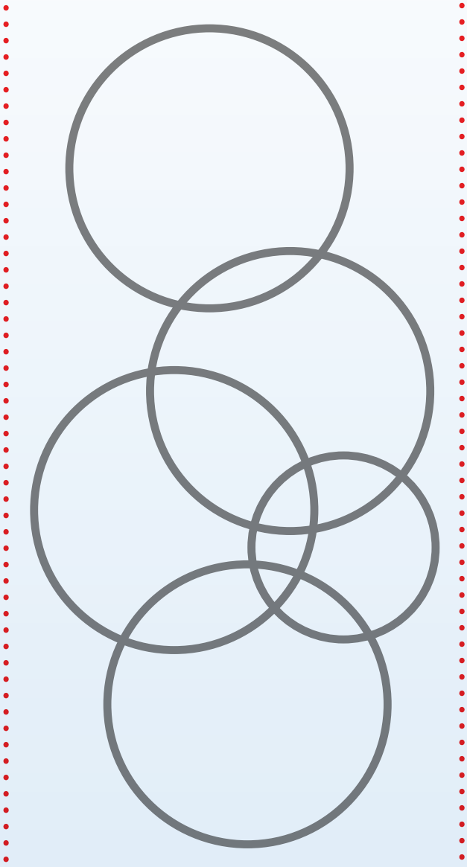
```
     IVec dir = pt2.diff(pt1);
      if (IRandom.percent(40)) { //bend
        new LineAgent(pt2, dir.dup().rot(IG.zaxis, PI/3), surf);
      }
      if (IRandom.percent(40)) { //bend the other way
        new LineAgent(pt2, dir.dup().rot(IG.zaxis, -PI/3), surf);
      }
      if (IRandom.percent(80)) { //straight
        new LineAgent(pt2, dir.dup(), surf);
      }
    }
  }
}
```

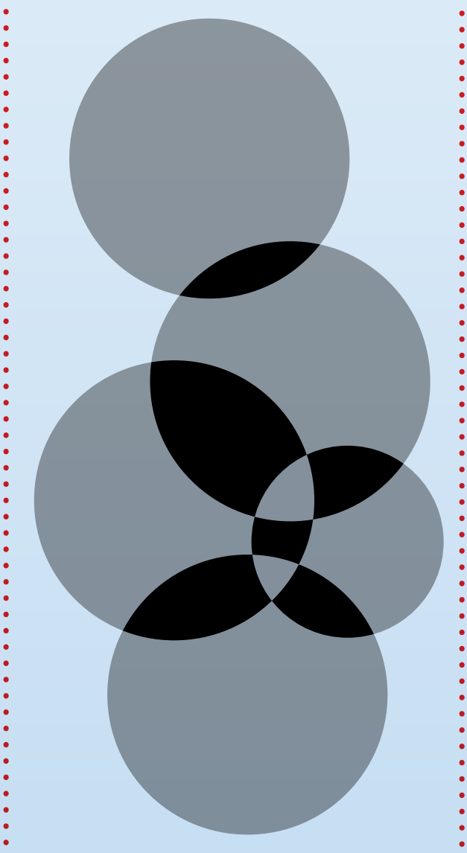




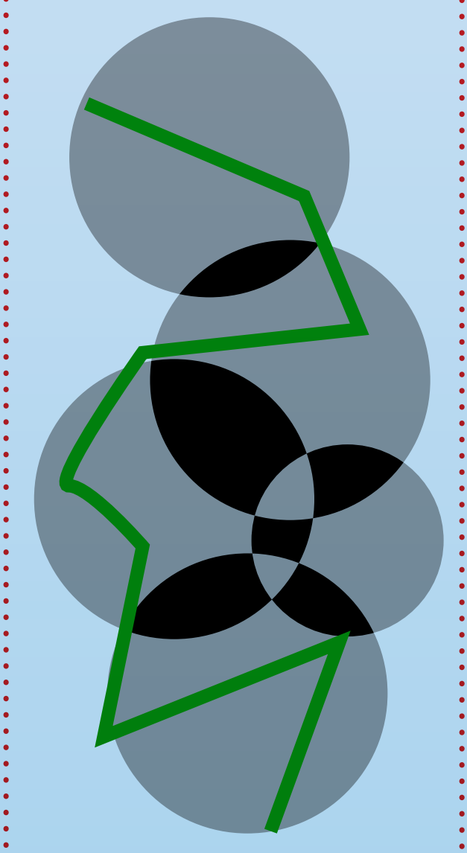
IG.duration(200);

```
i//interior plant growing inside boundary

class MyBoundary extends IAgent {
  IVec[] centers;
  double[] radii;

  MyBoundary() {
    centers = new IVec[5];
    radii = new double[5];

    centers[0] = new IVec(932.77, 1954.85, 2386.81);
    radii[0] = 195.85;

    centers[1] = new IVec(991.55, 2039.56, 2753.88);
    radii[1] = 319.91;

    centers[2] = new IVec(1066.77, 2343.17, 3245.16);
    radii[2] = 502.33;

    centers[3] = new IVec(1640.24, 2388.24, 2718.24);
    radii[3] = 355.54;

    centers[4] = new IVec(834.44, 2767.81, 3335.72);
    radii[4] = 411.77;
  }

  void interact(IDynamics agent) {
    if (agent instanceof LineAgent2) {
      LineAgent2 la = (LineAgent2)agent;

      boolean inside=false;
      for (int i=0; i<centers.length && !inside; i++) {
        double dist = la.pt2.dist(centers[i]);
        if (dist < radii[i]) {
          inside=true;
        }
      }

      if (!inside) {
        la.del();
      }
    }
  }
}
```





```
static class LineAgent2 extends IAgent {
  static double length = 33;
  static double clearance = 32.99; //less than length

  IVec pt1, pt2;
  boolean isColliding=false;

  LineAgent2(IVec pt, IVec dir) {
    pt1 = pt;
    pt2 = pt.dup().add(dir.dup().len(length));
  }

  void interact(IDynamics agent) {
    if (time == 0) { //only in the first time
      if (agent instanceof LineAgent2) {
        LineAgent2 lineAgent = (LineAgent2)agent;
        // checking clearance of end point
        if (lineAgent.pt2.dist(pt2) < clearance) {
          isColliding=true;

        }
      }
    }
  }
```



```
  void update() {
    if (isColliding) {
      del();
    }
    else if (time == 0) { //if not colliding
      new ICurve(pt1, pt2).clr(.3,.7,0.1);
      IVec dir = pt2.dif(pt1);

      //rotation axis with random direction
      IVec axis = IRandom.pt(-1, 1).len(1);

      if (IRandom.percent(50)) { //bend
        new LineAgent2(pt2, dir.dup().rot(axis,
        IRandom.get(PI/3, PI/3*2)));
      }
      if (IRandom.percent(50)) { //bend the other way
        new LineAgent2(pt2, dir.dup().rot(axis,
        -IRandom.get(PI/3, PI/3*2)));
      }
      if (IRandom.percent(99.9)) { //straight
        new LineAgent2(pt2, dir.dup());
      }
    }
  }
}
```