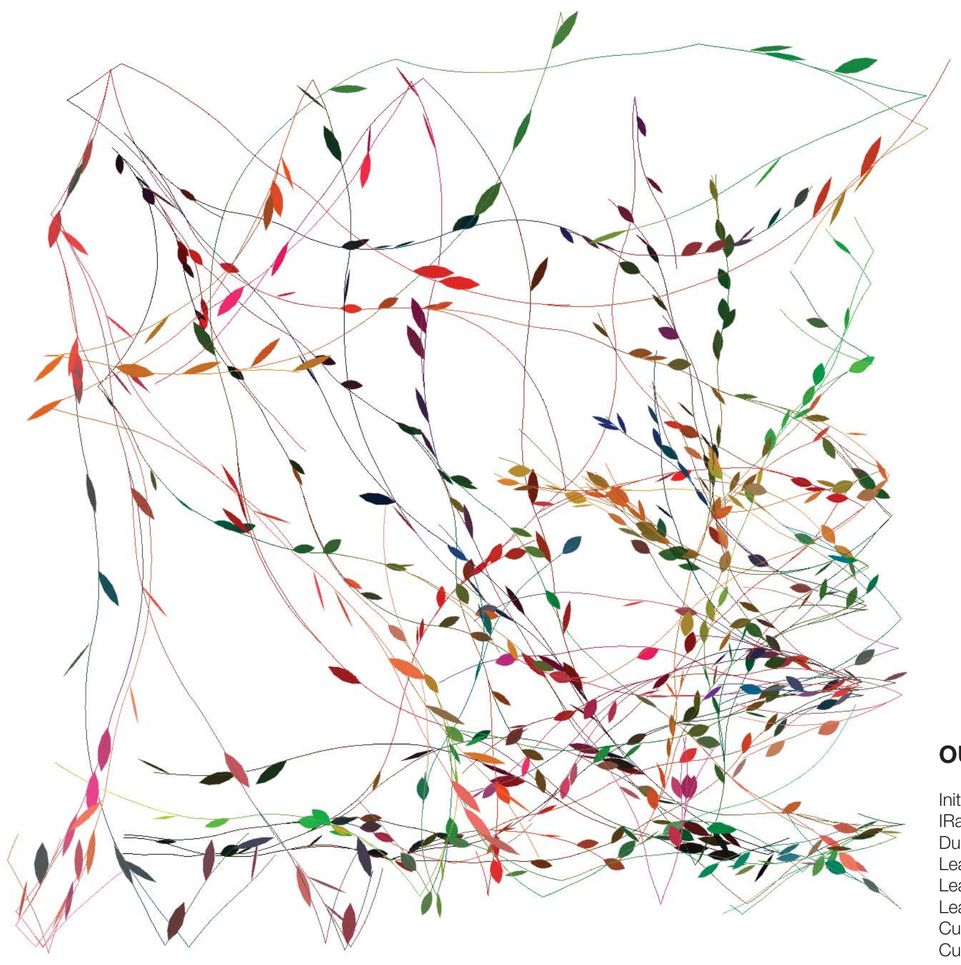
OUTPUT 1:

Initial Branching Number 3
IRandom Initial 4
Duration 160
Leaf Braching 30%
Leaf Rotate 80%
Leaf Rotate Angle -120° to 120°
Curve Rotate -12° to 12°
Curve Left to RIght Swap 5%

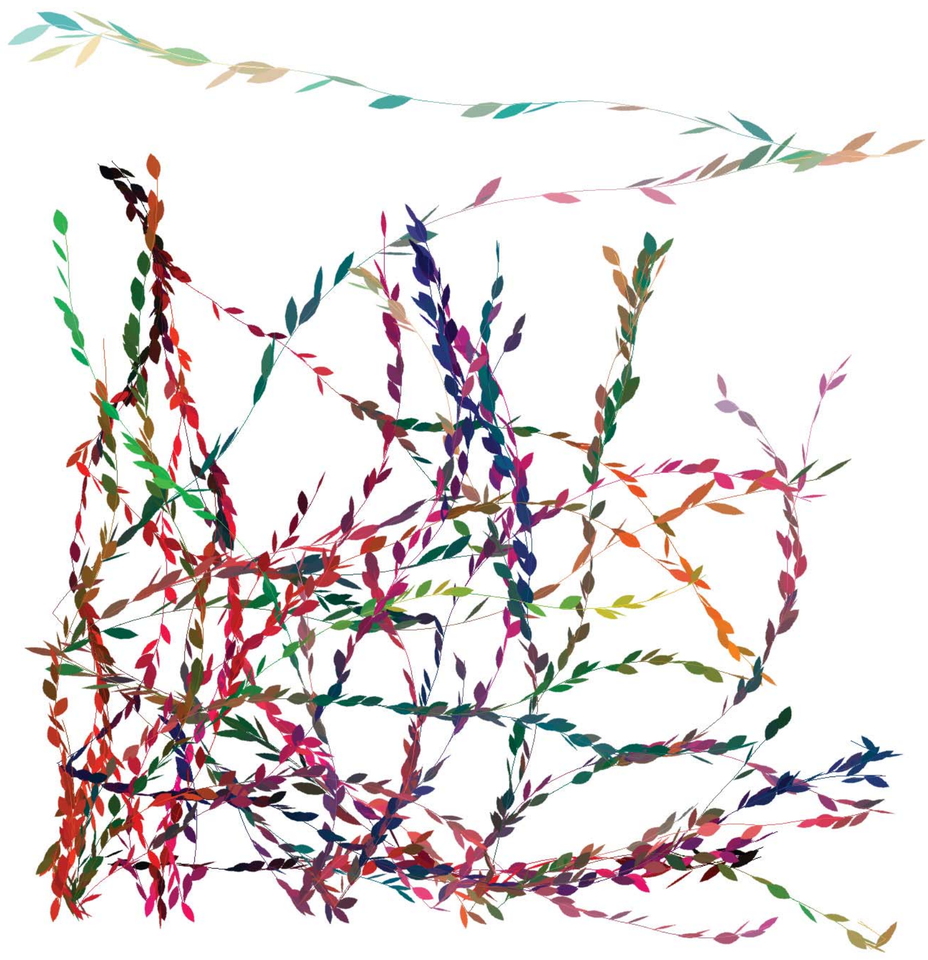
OUTPUT 2:

Initial Branching Number 2
IRandom Initial 5
Duration 250
Leaf Braching 20%
Leaf Rotate 70%
Leaf Rotate Angle -120° to 120°
Curve Rotate -12° to 12°
Curve Left to RIght Swap 5%



OUTPUT 3:

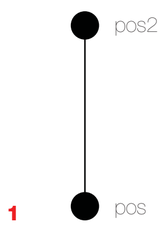
Initial Branching Number 3
IRandom Initial 4
Duration 160
Leaf Braching 10%
Leaf Rotate 85%
Leaf Rotate Angle-72° to 72°
Curve Rotate -10° to 10°
Curve Left to RIght Swap 10%



OUTPUT 4:

Initial Branching Number 2
IRandom Initial 5
Duration 200
Leaf Braching 50%
Leaf Rotate 80%
Leaf Rotate Angle -72° to 72°
Curve Rotate -9° to 9°
Curve Left to RIght Swap 10%

Agent Algorithm

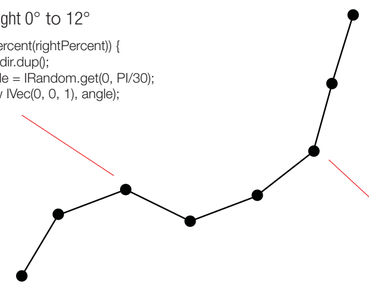


1

Line Agent

2 Rotate Angle Right 0° to 12°

```
if (IRandom.percent(rightPercent)) {
    IVec dir2 = dir.dup();
    double angle = IRandom.get(0, PI/30);
    dir2.rot(new IVec(0, 0, 1), angle);
```



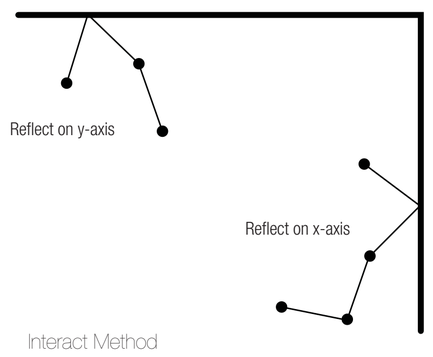
Update Method

3 Rotate Angle Left 0° to -12°

```
if (IRandom.percent(leftPercent)) {
    IVec dir2 = dir.dup();
    double angle = IRandom.get(-PI/30, 0);
    dir2.rot(new IVec(0, 0, 1), angle);
```

4 Left to Right Swap Chance 5%

```
if (IRandom.percent(5)) {
  double swap = leftPercent;
  leftPercent = rightPercent;
  rightPercent = swap;
  }
```



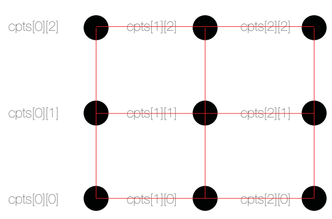
Interact Method

5 Check if next position is out of boundary

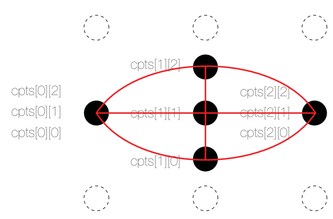
```
void interact( ArrayList < IDynamics > agents ) {
    for (int i=0; i < agents.size(); i++) {
      if (agents.get(i) instanceof MyBoundary) {
        MyBoundary boundary = (MyBoundary)agents.get(i);

        IVec nextPos = pos.cp(dir);
        if (nextPos.x < boundary.minx) {
          dir.ref(IG.xaxis);
        }
        else if (nextPos.x > boundary.maxx) {
          dir.ref(IG.xaxis);
        }
        if (nextPos.y < boundary.miny) {
          dir.ref(IG.yaxis);
        }
        else if (nextPos.y > boundary.maxy) {
          dir.ref(IG.yaxis);
        }
```

Agent Geometry



IVec Grid



Leaf Surface

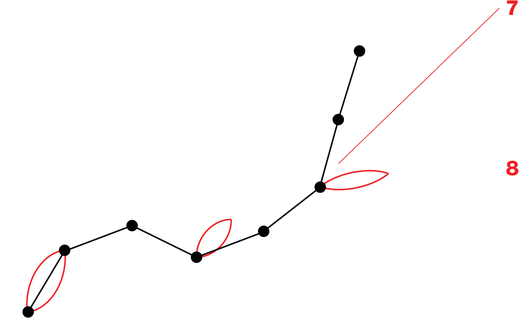
6 Leaf Branching 30%

```
if (IRandom.percent(30)) {

    double leafWidth = IRandom.getInt(1, 5);

    IVec dir3 = dir.dup().rot(IVec.zaxis, PI/2);
    dir3.len(leafWidth);

    IVec[][] cpts = new IVec[3][3];
    cpts[0][0] = pos.mid(pos2);
    cpts[0][1] = pos.mid(pos2);
    cpts[0][2] = pos.mid(pos2);
    cpts[1][0] = pos2.dup().add(dir3);
    cpts[1][1] = pos2.dup();
    cpts[1][2] = pos2.dup().sub(dir3);
    cpts[2][0] = pos2.mid(pos2.cp(dir));
    cpts[2][1] = pos2.mid(pos2.cp(dir));
    cpts[2][2] = pos2.mid(pos2.cp(dir));
```



7 Leaf Rotate Angle -120° to 120°

```
ISurface leafSurf = new ISurface(cpts, 2, 2). clr(r, g, b);

    double rotangel = IRandom.get(-PI/3, 0);
    double rotangel2 = IRandom.get(0, PI/3);
```

8 Leaft Rotate Chance 80%

```
if (IRandom.percent(80)) {
    leafSurf.rot(pos.mid(pos2), IVec.zaxis, rotangel2);
  }
  else {
    leafSurf .rot(pos.mid(pos2), IVec.zaxis, rotangel);
  }
  leafSurf.scale(pos.mid(pos2), 3);
}
```

```
import processing.opengl.*;
import igeo.*;

void setup() {
  size(680, 560, IG.GL);
  IRandom.init(4);
  IG.duration(160);


  new MyBoundary(0, 0, 1000, 1000);
  int num = 3;
  for (int i=0; i < num; i++) {
    new MyLineAgent(IRandom.pt(30, 30, 0),
    new IVec(10, 0, 0), new IVec(0, 0, 1), 99, 2).clr(0);
  }

  for (int i=0; i < num; i++) {
    new MyLineAgent(IRandom.pt(0, 950, 0, 30, 980, 0),
    new IVec(7, -7, 0), new IVec(0, 0, 1), 99, 2).clr(0);
  }

  for (int i=0; i < num; i++) {
    new MyLineAgent(IRandom.pt(900, 000, 0, 930, 30, 0),
    new IVec(-15, -5, 0), new IVec(0, 0, 1), 99, 2).clr(0);
  }
}

class MyBoundary extends IAgent {
  double minx, maxx, miny, maxy;
  MyBoundary(double x1, double y1, double x2, double y2) {
    minx = x1;
    miny = y1;
    maxx = x2;
    maxy = y2;
    IG.rect(new IVec(minx, miny, 0), maxx-minx, maxy-miny);
  }
}
```

1

```
static class MyLineAgent extends IAgent {
  IVec pos;
  IVec dir;
  IVec axis;
  double leftPercent, rightPercent;

  MyLineAgent(IVec pt, IVec dir, IVec ax, double leftPcnt, double
rightPcnt) {
    pos = pt;
    this.dir = dir;
    axis = ax;
    leftPercent = leftPcnt;
    rightPercent = rightPcnt;
  }
```

5

```
void interact( ArrayList < IDynamics > agents ) {
  for (int i=0; i < agents.size(); i++) {
    if (agents.get(i) instanceof MyBoundary) {
      MyBoundary boundary = (MyBoundary)agents.get(i);

      IVec nextPos = pos.cp(dir);
      if (nextPos.x < boundary.minx) {
        dir.ref(IG.xaxis);
      }
      else if (nextPos.x > boundary.maxx) {
        dir.ref(IG.xaxis);
      }
      if (nextPos.y < boundary.miny) {
        dir.ref(IG.yaxis);
      }
      else if (nextPos.y > boundary.maxy) {
        dir.ref(IG.yaxis);
      }
    }
  }
}
```

```
public void update() {
  super.update();
  IVec pos2 = pos.dup().add(dir);
  new ICurve(pos, pos2).clr(clr());
  IVec nextAxis1 = axis.dup();
```

6

```
if (IRandom.percent(30)) {

    double leafWidth = IRandom.getInt(1, 5);

    IVec dir3 = dir.dup().rot(IVec.zaxis, PI/2);
    dir3.len(leafWidth);

    IVec[][] cpts = new IVec[3][3];
    cpts[0][0] = pos.mid(pos2);
    cpts[0][1] = pos.mid(pos2);
    cpts[0][2] = pos.mid(pos2);
    cpts[1][0] = pos2.dup().add(dir3);
    cpts[1][1] = pos2.dup();
    cpts[1][2] = pos2.dup().sub(dir3);
    cpts[2][0] = pos2.mid(pos2.cp(dir));
    cpts[2][1] = pos2.mid(pos2.cp(dir));
    cpts[2][2] = pos2.mid(pos2.cp(dir));
```

```
    int r = clr().getRed() + IRandom.getInt(-40, 40);
    int g = clr().getGreen() + IRandom.getInt(-10, 10);
    int b = clr().getBlue() + IRandom.getInt(-10, 10);


    ISurface leafSurf = new ISurface(cpts, 2, 2). clr(r, g, b);
```

7

```
    double rotangel = IRandom.get(-PI/3, 0);
    double rotangel2 = IRandom.get(0, PI/3);
```

8

```
    if (IRandom.percent(80)) {
      leafSurf.rot(pos.mid(pos2), IVec.zaxis, rotangel2);
    }
    else {
      leafSurf .rot(pos.mid(pos2), IVec.zaxis, rotangel);
    }
    leafSurf.scale(pos.mid(pos2), 3);
  }
```

4

```
  if (IRandom.percent(5)) {
    double swap = leftPercent;
    leftPercent = rightPercent;
    rightPercent = swap;
  }
```

3

```
  if (IRandom.percent(leftPercent)) {
    IVec dir2 = dir.dup();

    double angle = IRandom.get(-PI/30, 0);


    dir2.rot(new IVec(0, 0, 1), angle);


    int r = clr().getRed() + IRandom.getInt(-50, 50);
    int g = clr().getGreen() + IRandom.getInt(-10, 10);
    int b = clr().getBlue() + IRandom.getInt(-10, 10);
```

```
      new MyLineAgent(pos2, dir2, nextAxis1, leftPercent,
  rightPercent).clr(r, g, b);
    }

    del();
  }
}
```

2

```
  if (IRandom.percent(rightPercent)) {
    IVec dir2 = dir.dup();


    double angle = IRandom.get(0, PI/30);


    dir2.rot(new IVec(0, 0, 1), angle);


    int r = clr().getRed() + IRandom.getInt(-50, 50);
    int g = clr().getGreen() + IRandom.getInt(-10, 10);
    int b = clr().getBlue() + IRandom.getInt(-10, 10);
```

```
      new MyLineAgent(pos2, dir2, nextAxis1, leftPercent,
  rightPercent).clr(r, g, b);
    }
```