

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

```
void setup(){
  size(1000, 1000, IG.GL);
  IRand.init(3);
  IG.duration(50);
  new LineAgent(new IVec(0,0,0), new IVec(1,0,0)).clr(0);
  IG.fill();
}
```

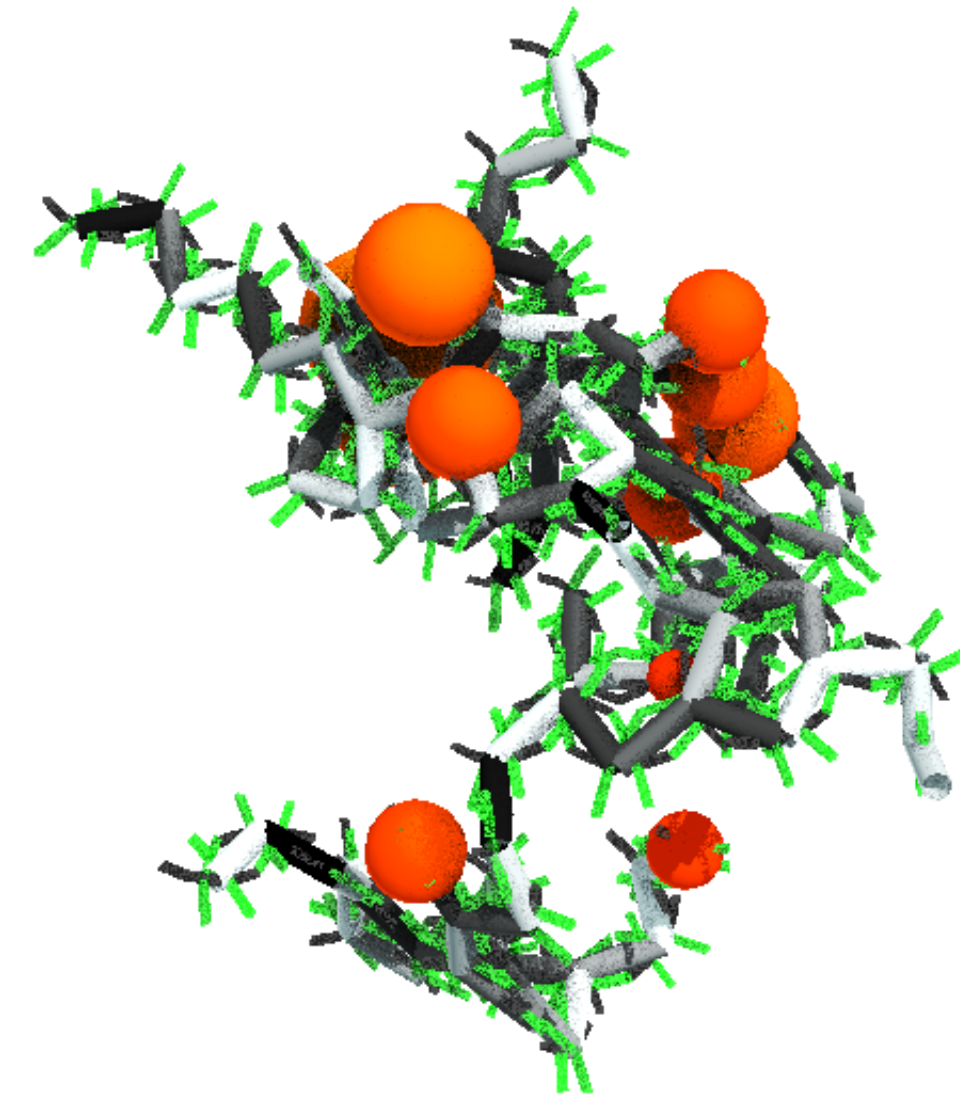
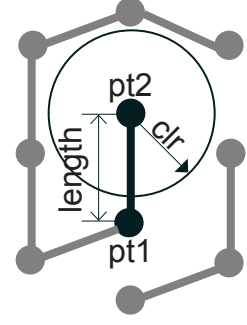
RUNNING TIME

```
static class LineAgent extends IAgent{
  static double length = 10.5;
  static double clearance = 5.5; //less than length
  IVec pt1, pt2;
  boolean isColliding=false;

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



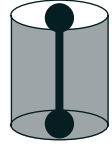
```
void interact(ArrayList < IDynamics > agents){
  super.interact(agents);
  if(time == 0){ //only in the first time
    for(int i=0; i < agents.size() && !isColliding; i++){
      if(agents.get(i) instanceof LineAgent){
        LineAgent lineAgent =
          (LineAgent)agents.get(i);
        if(lineAgent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



```
void update(){
  super.update();

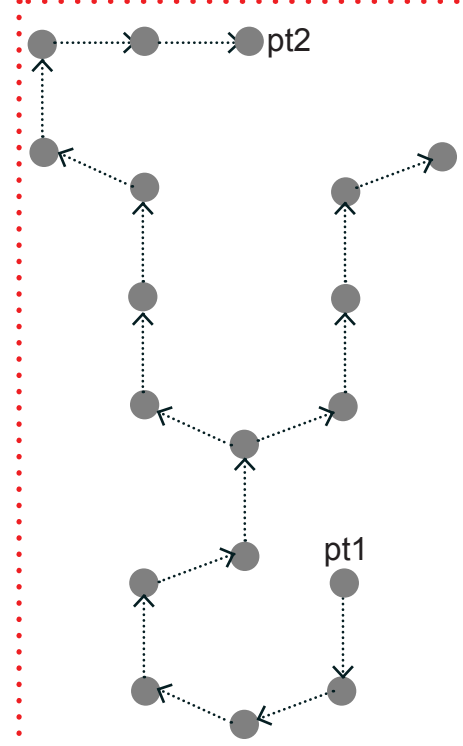
  if(isColliding){ del(); }
  else if(time == 0){ //if not colliding

    ICurve structure= new ICurve(pt1,pt2); // center line
    IG.pipe(structure,1.70).clr(IRandom.gray());
  }
}
```



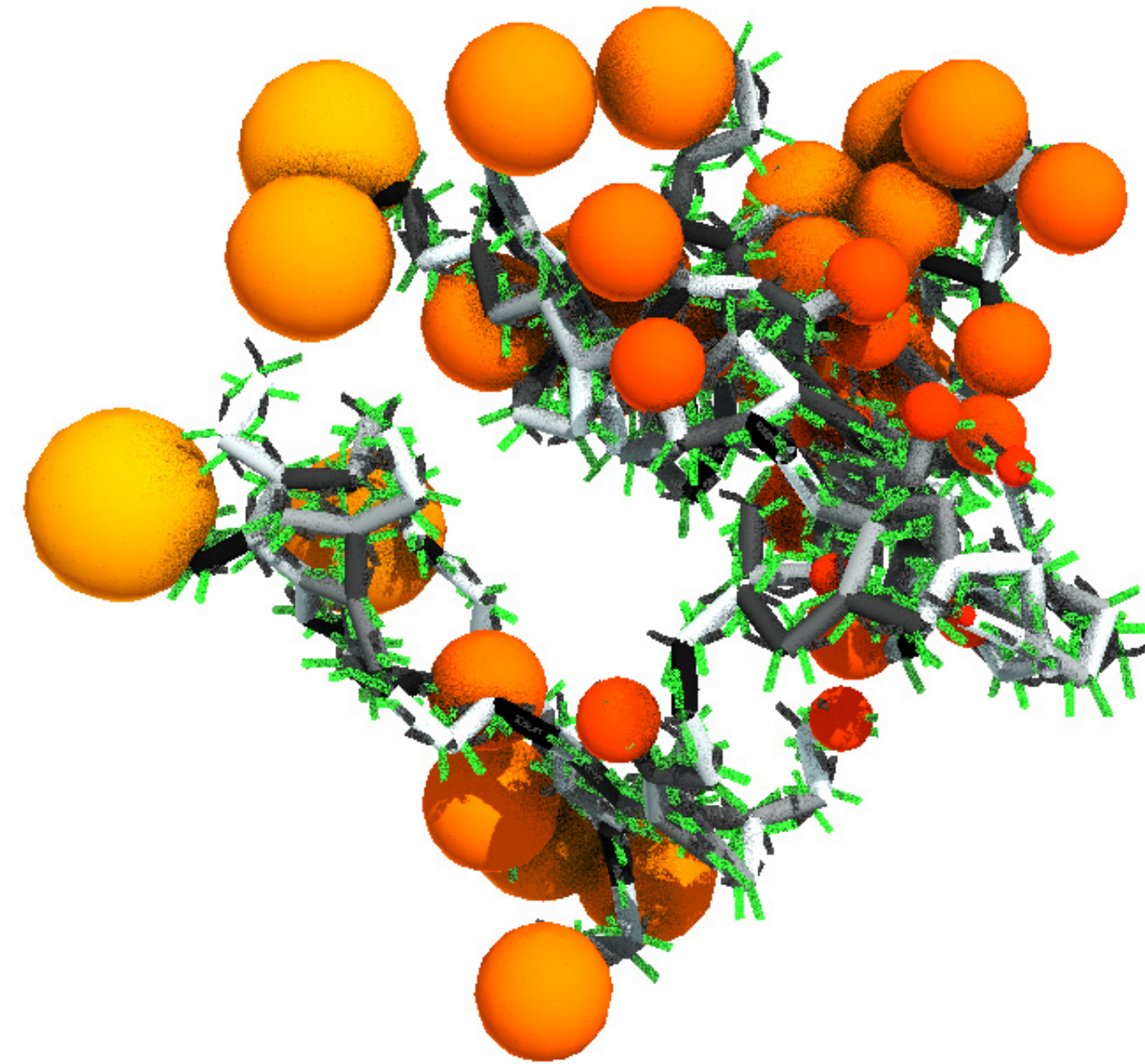
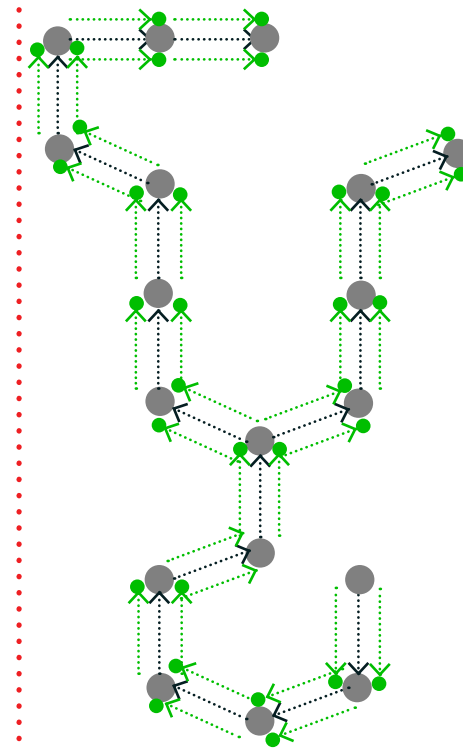
IG.duration(20);

```
IVec dir = pt2.diff(pt1);
//making axis perpendicular to dir
IVec axis = IRand.pt(0,1).cross(dir);
// child dir & points
IVec nextDir1 = dir.dup().rot(axis,-PI/2);
IVec nextDir2 = dir.dup().rot(axis,PI/3);
IVec nextPt1 = pt2.cp(nextDir1);
IVec nextPt2 = pt2.cp(nextDir2);
//midpoints
IVec mid1 = pt1.mid(pt2);
IVec mid2 = pt2.mid(nextPt1);
IVec mid3 = pt2.mid(nextPt2);
//mid of midpoints
IVec quarter1 = pt2.mid(mid1);
IVec quarter2 = pt2.mid(mid2);
IVec quarter3 = pt2.mid(mid3);
```



```
double offsetWidth = -3;
IVec offset1 = dir.cross(axis).len(offsetWidth);
IVec offset2 = nextDir1.cross(axis).len(offsetWidth);
IVec offset3 = nextDir2.cross(axis).len(offsetWidth);

//offset edge points 1
IVec edgePt11 = mid1.cp(offset1);
IVec edgePt12 = quarter1.cp(offset1);
IVec edgePt13 = quarter2.cp(offset2);
IVec edgePt14 = mid2.cp(offset2);
//offset edge points 2
offset2.flip(); //offset to opposite
IVec edgePt21 = mid2.cp(offset2);
IVec edgePt22 = quarter2.cp(offset2);
IVec edgePt23 = quarter3.cp(offset3);
IVec edgePt24 = mid3.cp(offset3);
//offset edge points 3
offset1.flip(); //offset to opposite
offset3.flip(); //offset to opposite
IVec edgePt31 = mid3.cp(offset3);
IVec edgePt32 = quarter3.cp(offset3);
IVec edgePt33 = quarter1.cp(offset1);
IVec edgePt34 = mid1.cp(offset1);
```



IG.duration(30);

```
//degree 3 curves
new ICurve(new IVec[]{ edgePt11,edgePt12,
  edgePt13,edgePt14 }, 3).clr(1,0,0);

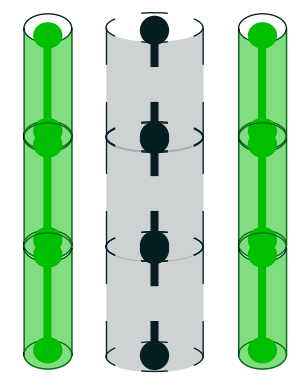
ICurve structure1 = new ICurve(new IVec[]{ edgePt11,edgePt12,
  edgePt13,edgePt14 }, 3).clr(1,0,0); // center line
IG.squarePipe(structure1,1).clr(0,0,0,5);

new ICurve(new IVec[]{ edgePt21,edgePt22,
  edgePt23,edgePt24 }, 3).clr(0,1,1,1);

ICurve structure2 = new ICurve(new IVec[]{ edgePt21,edgePt22,
  edgePt23,edgePt24 }, 3).clr(0,1,1,1);
IG.squarePipe(structure2,1).clr(0,75,0,5);

new ICurve(new IVec[]{ edgePt31,edgePt32,
  edgePt33,edgePt34 }, 3).clr(1,1,1,1);

ICurve structure3 = new ICurve(new IVec[]{ edgePt31,edgePt32,
  edgePt33,edgePt34 }, 3).clr(1,1,1,1);
IG.squarePipe(structure3,1).clr(0,5,0,5);
```

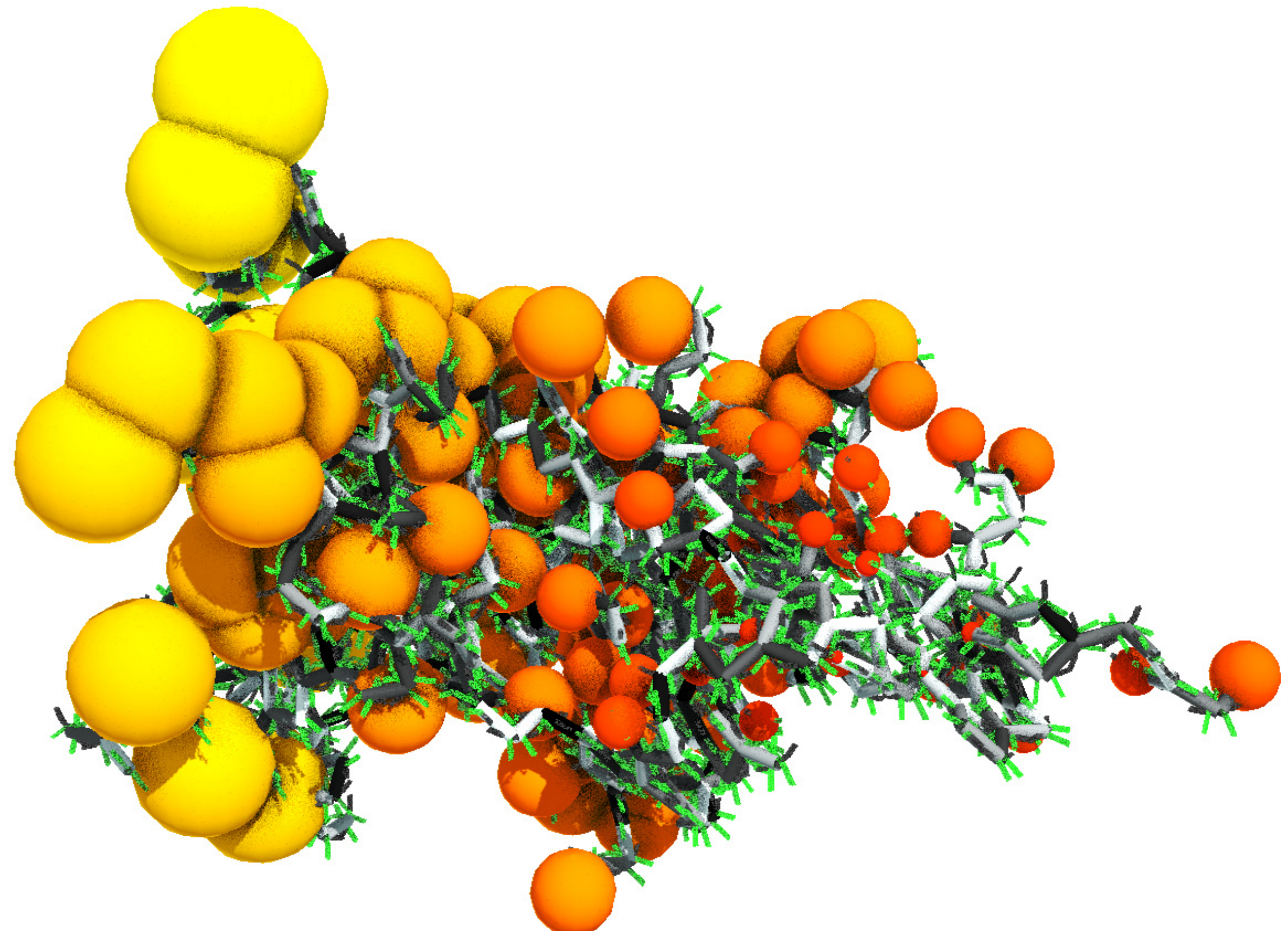
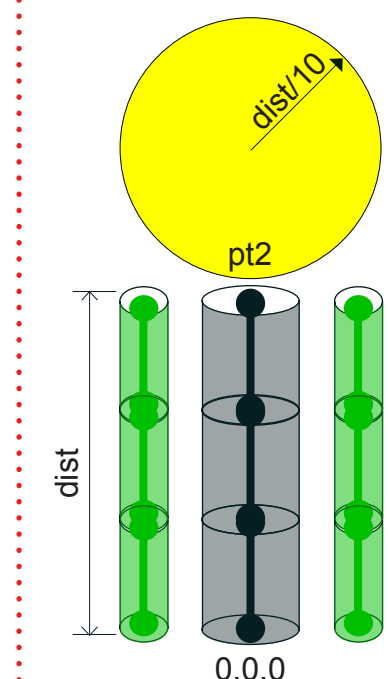


```
boolean anyChild=false;

if(IRand.percent(70)){ //bend
  new LineAgent(pt2, nextDir1);
  anyChild=true;
}

if(IRand.percent(50)){ //bend the other way
  new LineAgent(pt2, nextDir2);
  anyChild=true;
}

if(!anyChild || IG.time()==50){
  double dist = pt2.dist(new IVec(0,0,0));
  double radius = dist/10;
  new ISphere(pt2,radius ).clr(dist*150,dist/250,0);
}
}
```



IG.duration(50);