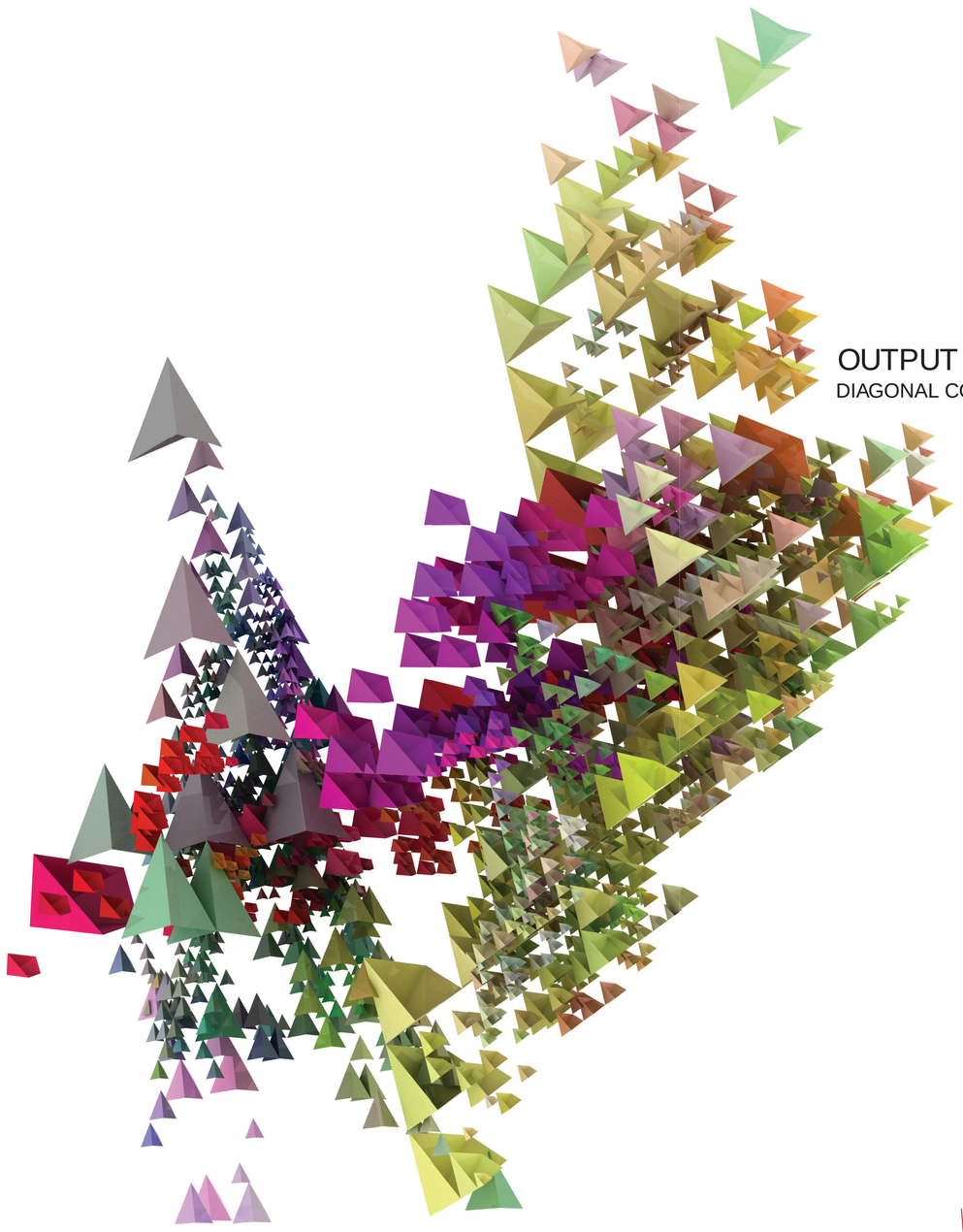
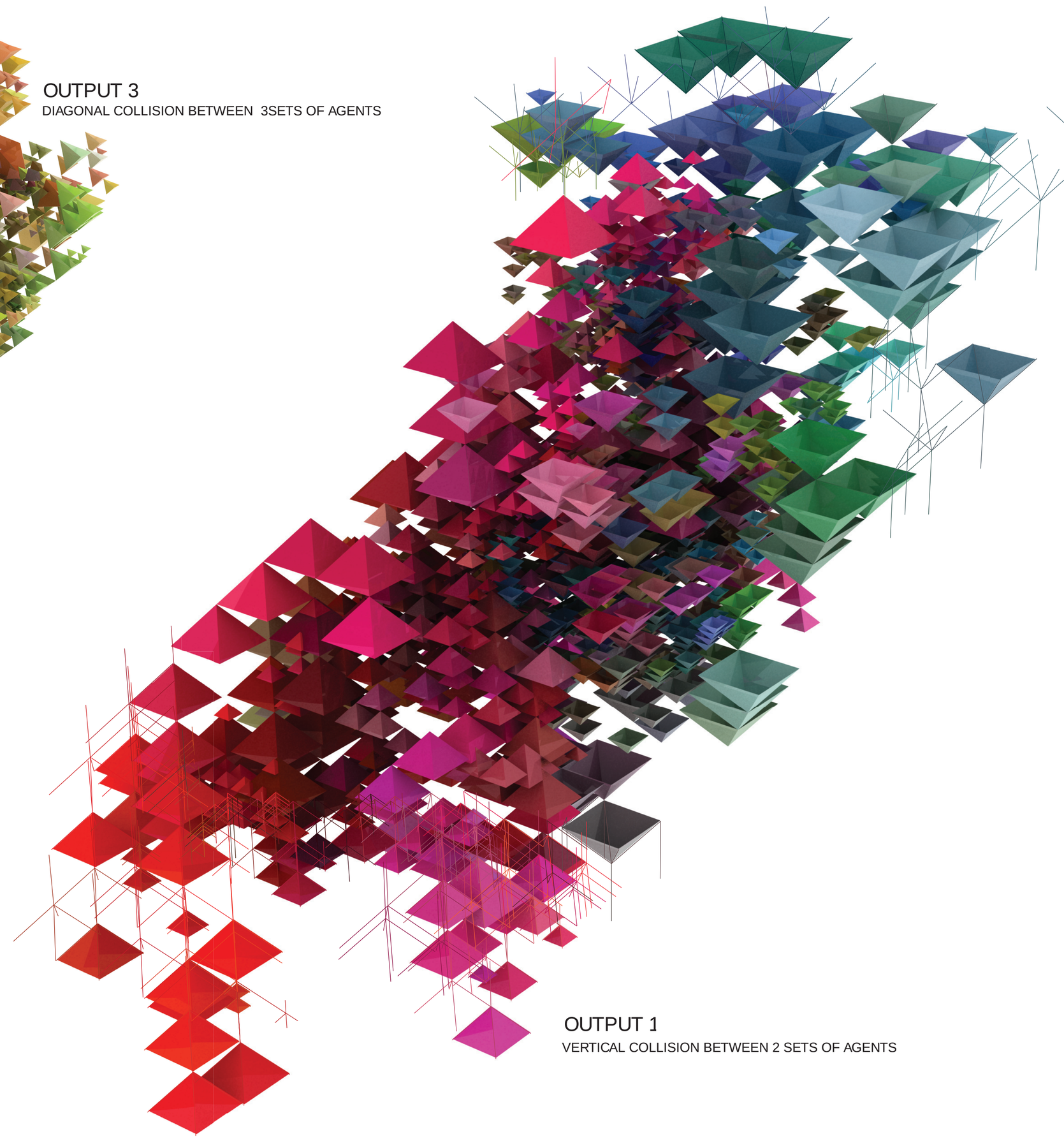
AGENTS IN COLLISION



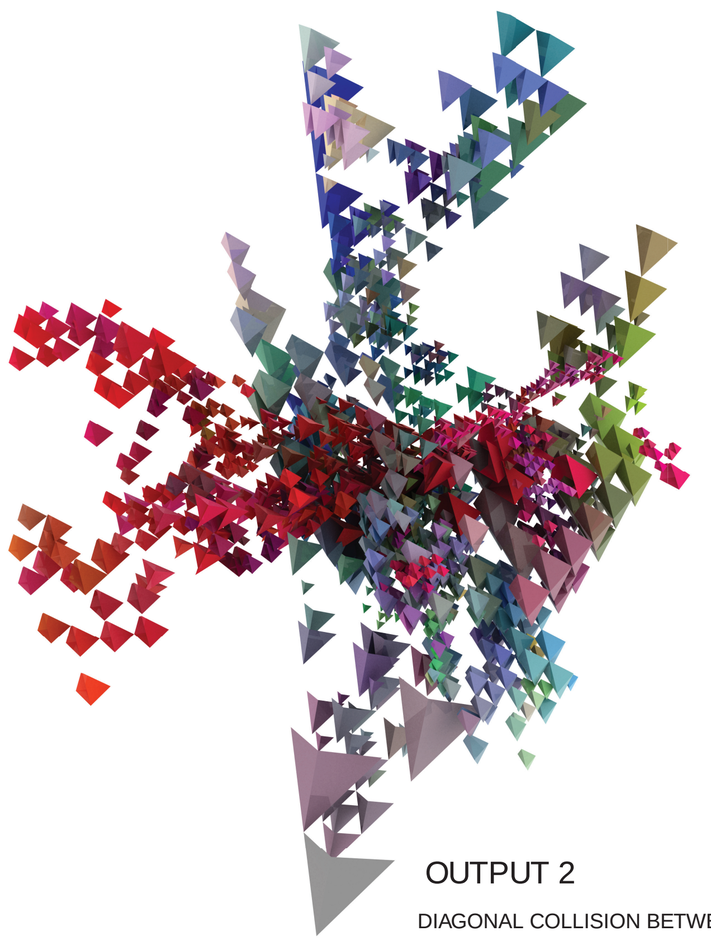
OUTPUT 3

DIAGONAL COLLISION BETWEEN 3SETS OF AGENTS



OUTPUT 1

VERTICAL COLLISION BETWEEN 2 SETS OF AGENTS

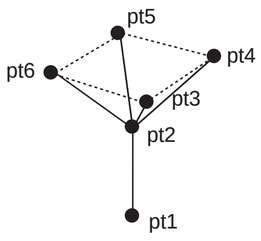


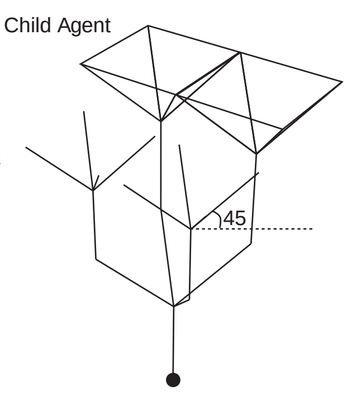
OUTPUT 2

DIAGONAL COLLISION BETWEEN 2 SETS OF AGENTS

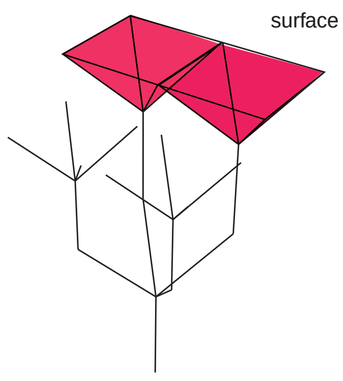
AGENT GEOMETRY+ ALGORITHM DIAGRAM

Line Agent

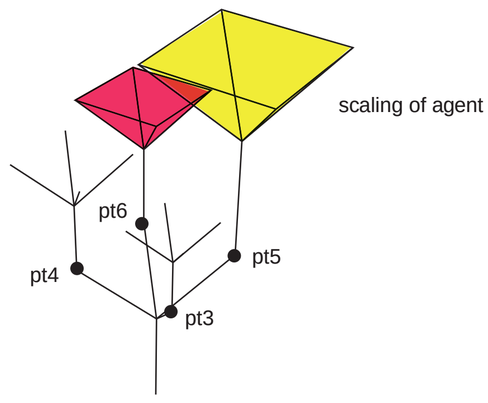




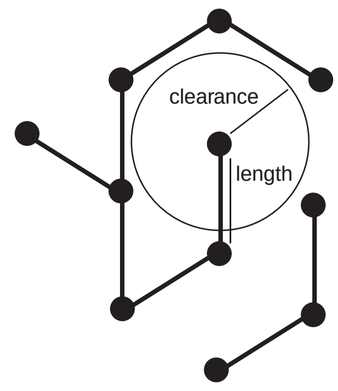
update method



50% chance of forming a surface



50% chance that lineagent starting at pt5 will be scaled between 1-5,
50% chance that lineagent starting at pt,3,4,6 will be scaled down by 0.99



An agent checks alll other existing agents and if it finds any pt2 of exisiting agents within clearance distance it deletes itself

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

void setup(){
  size(600, 600, IG.GL);
  IRandom.init(5);
  IG.duration(18);

    new LineAgent(new IVec(30,-30,30), new IVec(5,-5,5)).clr(0.5,0.5);
    new LineAgent(new IVec(0,0,0), new IVec(5,5,-5)).clr(1.0,0,0.5);
    new LineAgent(new IVec(-30,-30,0), new IVec(0,5,-5)).clr(1.0,1.0,0.5);
}

static class LineAgent extends IAgent{
  //static double length = 10;
  //static double clearance = 9.99; //less than length
  static IVec axis1 = new IVec(10, 10, 10);
  static IVec axis2 = new IVec(10, -10, 10);
  static IVec axis3 = new IVec(-10, -10, 10);
  static IVec axis4 = new IVec(-10, 10, 10);
  static IVec ctr= new IVec(0, 0, 10);

  IVec pt1, pt2;
  boolean isColliding=false;

  IVec pt3;
  IVec pt4;
  IVec pt5;
  IVec pt6;
```

```
  LineAgent(IVec pt, IVec dir){
    pt1 = pt;
    pt2 = pt.dup().add(dir);
   // pt3 = pt.dup().add(dir.dup().rot(axis1.dup() , PI/3));
    //pt4 = pt.dup().add(dir.dup().rot(axis2.dup() , PI/3));
   // pt5 = pt.dup().add(dir.dup().rot(axis3.dup() , PI/3));
   // pt6 = pt.dup().add(dir.dup().rot(axis4.dup() , PI/3));
    pt3 = pt.dup().add(dir.dup().flip().rot(axis1.dup() , PI/3));
    pt4 = pt.dup().add(dir.dup().flip().rot(axis2.dup() , PI/3));
    pt5 = pt.dup().add(dir.dup().flip().rot(axis3.dup() , PI/3));
    pt6 = pt.dup().add(dir.dup().flip().rot(axis4.dup() , PI/3));
  }

  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) < pt1.dist(pt2)*0.5){
              isColliding=true;
            }
          }
        }
      }
    }
  }
```

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

  if(isColliding){
    del();
  }
  else if(time == 0){ //if not colliding
    new ICurve(pt2,pt1).clr(this.clr());
    new ICurve(pt1,pt3).clr(this.clr());
    new ICurve(pt1,pt4).clr(this.clr());
    new ICurve(pt1,pt5).clr(this.clr());
    new ICurve(pt1,pt6).clr(this.clr());

    IVec[][] cpts = new IVec[2][4];
    //cpts[0][0] = pt2;
    //cpts[0][1] = pt2;
    //cpts[0][2] = pt2;
    //cpts[0][3] = pt2;
    cpts[0][0] = pt1;
    cpts[0][1] = pt1;
    cpts[0][2] = pt1;
    cpts[0][3] = pt1;
    cpts[1][0] = pt3;
    cpts[1][1] = pt4;
    cpts[1][2] = pt5;
    cpts[1][3] = pt6;
```

```
      //ISurface surface1 = new ISurface(cpts, 1, 1, false, true)
     // surface1.scale(new IVec(ctr.dup()),0.8);
      int r = clr().getRed() + IRandom.getInt(-30, 30);
      int g = clr().getGreen() + IRandom.getInt(-30, 30);
      int b = clr().getBlue() + IRandom.getInt(-30, 30);

        IVec dir = pt2.diff(pt1);
      if(IRandom.percent(50)){
        ISurface surface1 = new ISurface(cpts, 1, 1, false,
true).clr(this.clr());
      }
       if(IRandom.percent(50)){ //bend

      new LineAgent(pt5, dir.dup().len(IRandom.get(1, 5))).clr(r,
g, b);
         //new LineAgent(pt3, dir.dup().mul(0.99));
       }

       if(IRandom.percent(50)){  //bend
         new LineAgent(pt6, dir.dup().mul(0.99)).clr(r, g, b);
       }

       if(IRandom.percent(50)){ //bend
         new LineAgent(pt4, dir.dup().mul(0.99)).clr(r, g, b);
    }
    }
   }
  }
```

CODING FORM 2011 FALL
SAMUEL LIEW
INSTRUCTOR: SATURO SUGIHARA