mass: 0.1
size: 64.0
repel: 25.0

创意编程

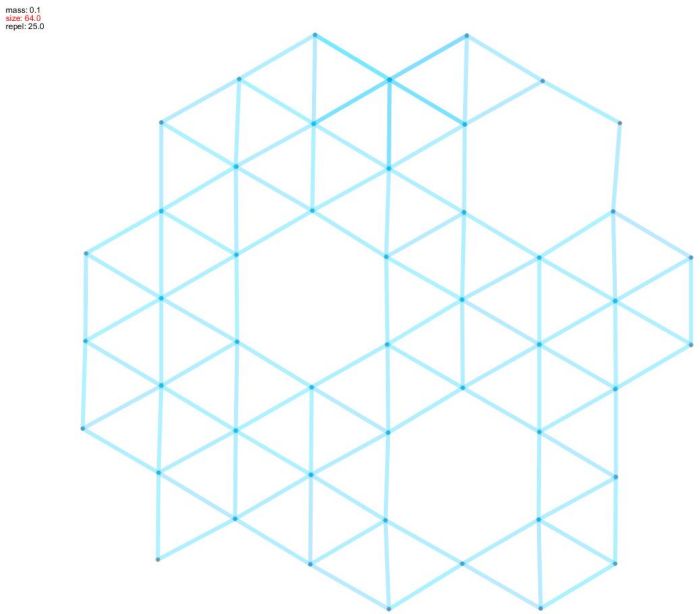
课后作业 (07-01) 粒子系统2

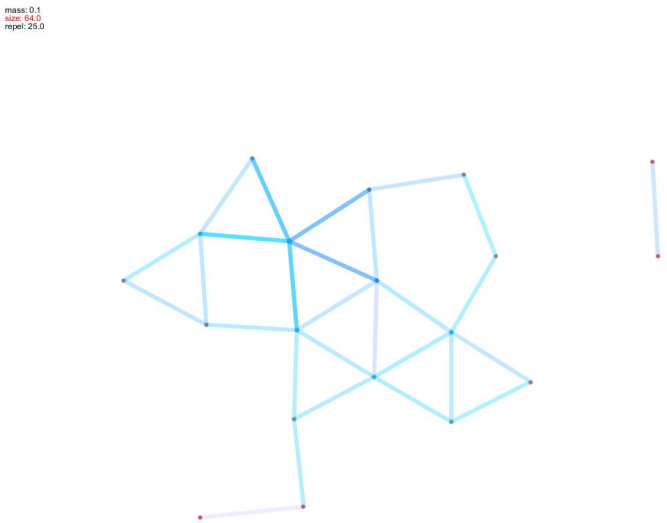
孔安梨 519430990004

功能介绍

鼠标点击

//左键增加粒子
//右键消除粒子

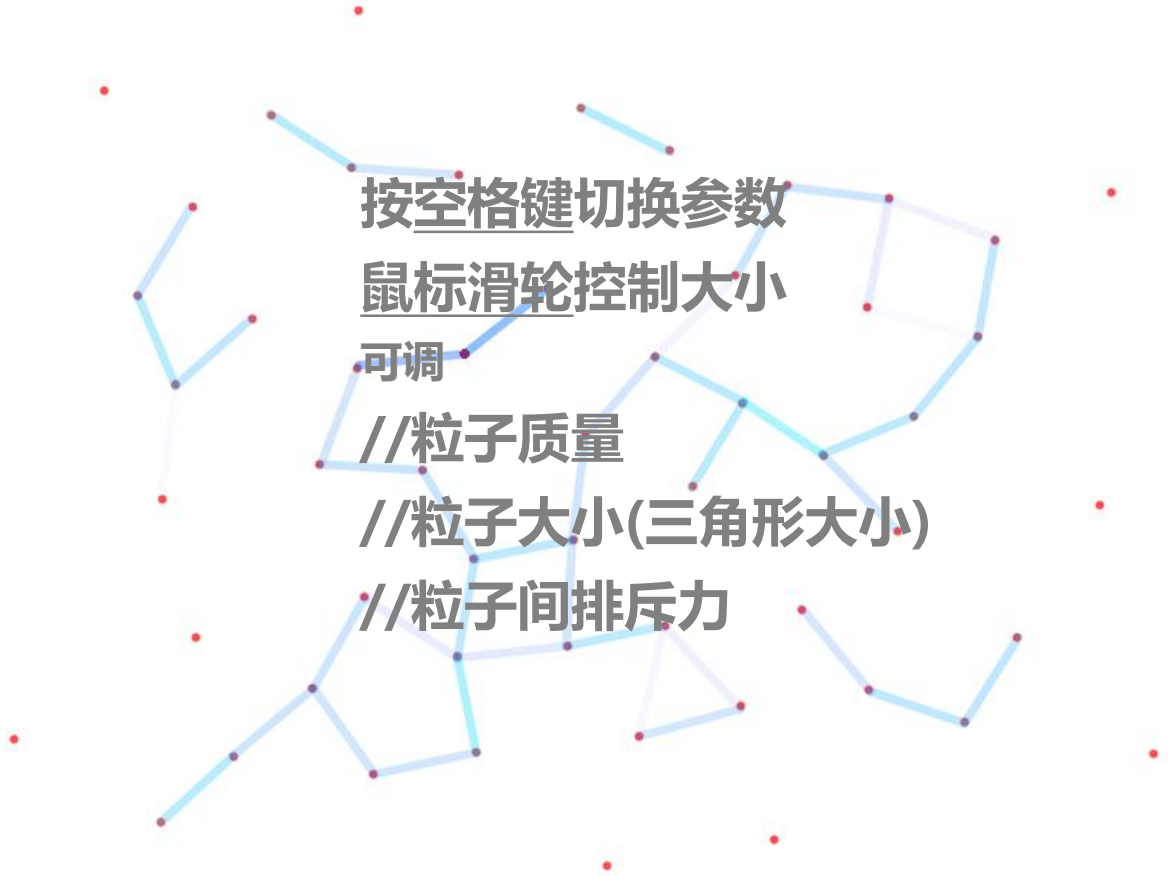




mass: 0.1
size: 33.0
repel: 74.0

功能介绍

参数显示和调整



按空格键切换参数

鼠标滑轮控制大小

可调

//粒子质量

//粒子大小(三角形大小)

//粒子间排斥力

功能介绍

mass: 0.1
size: 33.0
repel: 74.0

清除画面

键盘C键
//清空画面

结构设计说明

这次的系统根据上次的系统简化了可交互的选项(上一个系统可调参数过多)，对用户更加友好。这次的系统只展示了粒子的中心点，并根据粒子是否有交叉以及交叉距离的大小画线，稳定下来后产生相同大小的三角形。

全局变量有描边粗细，空气阻力，初始粒子数，粒子弹性，这里的size我在最后放入了用户可用鼠标滑轮调整的选项里。

```
1  //global variables, feel free to tweak
2  float size = 100;
3  float defaultStrokeWeight = 4;
4  float friction = 0.02;
5  int initBallCount = 50;
6  float elascity = 0.98;
7
8  //ajustables
9  float mass = 0.5;
10 float re = 50;
11 int tool = 0;
12
13
14 ArrayList<Ball> balls;
15
```

计算每个粒子的加速度:

计算每一个粒子所受另外所有粒子的影响的力所导致的加速度并计算其总和.

得出两个例子之间的向量以及距离

若两个粒子相撞，确保距离有一个最低限度，以保证加速度不会过大.
计算其他的排斥力、阻力等等。

根据距离与对方粒子的质量计算重力和加速度的强度，并加入加速度的总和

```
60  //update force
61  for (int i = 0; i < particles.size(); i++)
62  {
63    PVector a = new PVector(0,0);  //initialize accleration
64    Particle pi = particles.get(i);
65    for (int j = 0; j < particles.size(); j++)
66    {
67      if (i == j) continue;
68      Particle pj = particles.get(j);
69      PVector r = pj.loc.copy().sub(pi.loc);
70      float d = r.mag();
71      if (d < pi.rad + pj.rad)    //particles are touching/colliding
72      {
73        d = pi.rad + pj.rad;                    //ensure the gforce is not too large
74        particles.get(i).vel.mult(elascity);    //reduce velocity to create 'sticking' effect
75        //repel force between particles to create 'bouncing' effect
76        float mult2 = -1*re/ ((particles.get(i).mass+1)*d);
77        PVector aj2 = r.normalize().mult(mult2);
78        a.add(aj2);
79      }
80
81      float mult = gr*particles.get(j).mass / (d*d);  //magnitude of gforce
82      PVector aj = r.normalize().mult(mult);          //force vector
83      a.add(aj);
84    }
85    particles.get(i).acc = a;
86  }
87
```

在粒子之间画线

```
53        if (d < 2.4*size)
54        {
55          c = mapScalarToRGB(0.5*size,2.4*size,d);
56          //stroke(c,255*(1.2*startSize/d-0.5));
57          stroke(c,255-map(d,0,2.4*size,0,253));
58          line(balls.get(i).center.x,balls.get(i).center.y,balls.get(j).center.x,balls.get(j).center.y);
59        }
```

鼠标左键点击增加新粒子，右键点击清除当前位置的粒子
按下鼠标c键清除所有粒子

```
104 void mouseClicked()
105 {
106   if (mouseButton == LEFT) {
107     PVector cen = new PVector(mouseX,mouseY);
108     balls.add(new Ball(cen));
109   }
110   if (mouseButton == RIGHT) {
111     for (int i = balls.size()-1; i >= 0; i--) {
112       float d = dist(balls.get(i).center.x,balls.get(i).center.y,mouseX,mouseY);
113       if (d < 20) balls.remove(i);
114     }
115   }
116 }
117
118 void keyPressed() {
119   if (key == ' ') {
120     tool = (tool+1)%3;
121   }
122   if (key == 'c') {
123     balls.clear();
124   }
125 }
```

颜色和透明度根据粒子间的距离渐变

```
175 //utility functions
176 color mapScalarToRGB(float min, float max, float scalar)
177 {
178   //map size to rgb
179     float r = 255,g = 0,b = 0;
180     float a = 4*(scalar-min)/(max-min);
181     int X = floor(a);
182     float Y = 255*(a-X);
183     switch (X)
184     {
185       case 0: r=255;g=Y;b=0;break;
186       case 1: r=255-Y;g=255;b=0;break;
187       case 2: r=0;g=255;b=Y;break;
188       case 3: r=0;g=255-Y;b=255;break;
189       case 4: r=0;g=0;b=255;break;
190     }
191     return color(r,g,b);
192 }
```