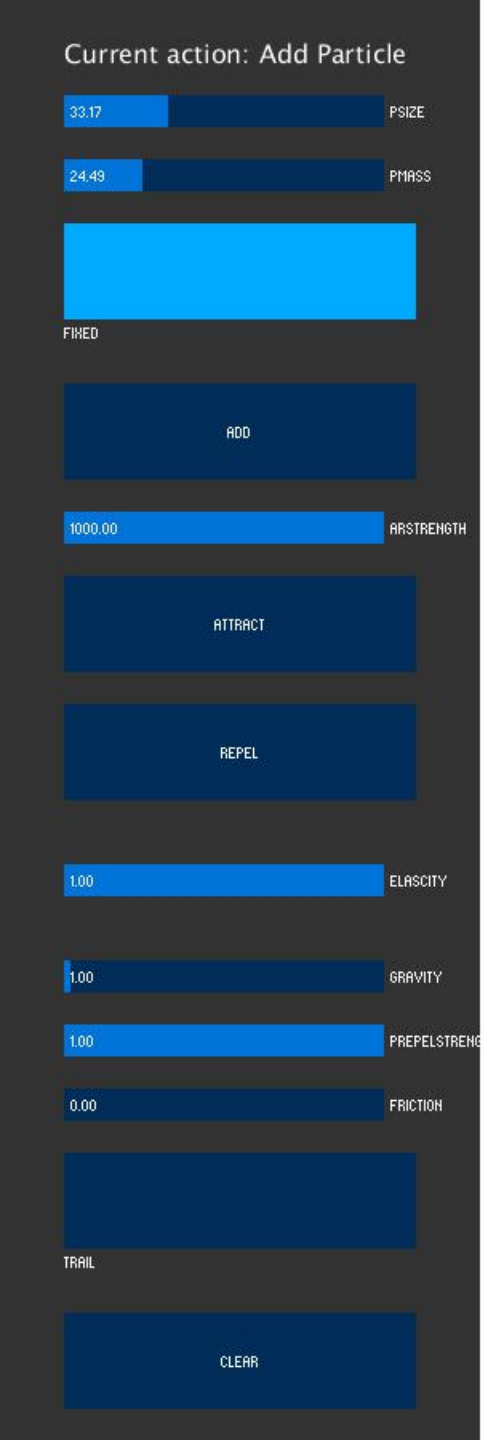

Current action: Add Particle
6.94
PSIZE
18.50
PMASS
FIXED
ADD
1000.00
ARSTRENGTH
ATTRACT
REPEL
0.54
ELASCITY
0.75
GRAVITY
1.00
PREPELSTRENG
0.00
FRICTION
TRAIL
CLEAR
创意编程
课后作业 (06-01) 粒子系统
孔安梨 519430990004


Current action: Add Particle
15.84
PSIZE
100.00
PMASS
FIXED
ADD
1000.00
ARSTRENGTH
ATTRACT
REPEL
0.56
ELASCITY
38.50
GRAVITY
0.00
PREPELSTRENG
0.00
FRICTION
TRAIL
CLEAR
可调
//粒子大小
//粒子质量
//动态/固态
功能介绍
粒子添加
//粒子轨道开关



功能介绍

参数模式

可调

//粒子弹性

//粒子间引力

//粒子间排斥力

//空气阻力

Current action: Add Particle

33.17 PSIZE

24.49 PMASS

FIXED

ADD

1000.00 ARSTRENGTH

ATTRACT

REPEL

1.00 ELASCITY

1.00 GRAVITY

1.00 PREPELSTREN

0.00 FRICTION

TRAIL

CLEAR

鼠标点击开关

//力的大小

//吸引力按钮

//排斥力按钮

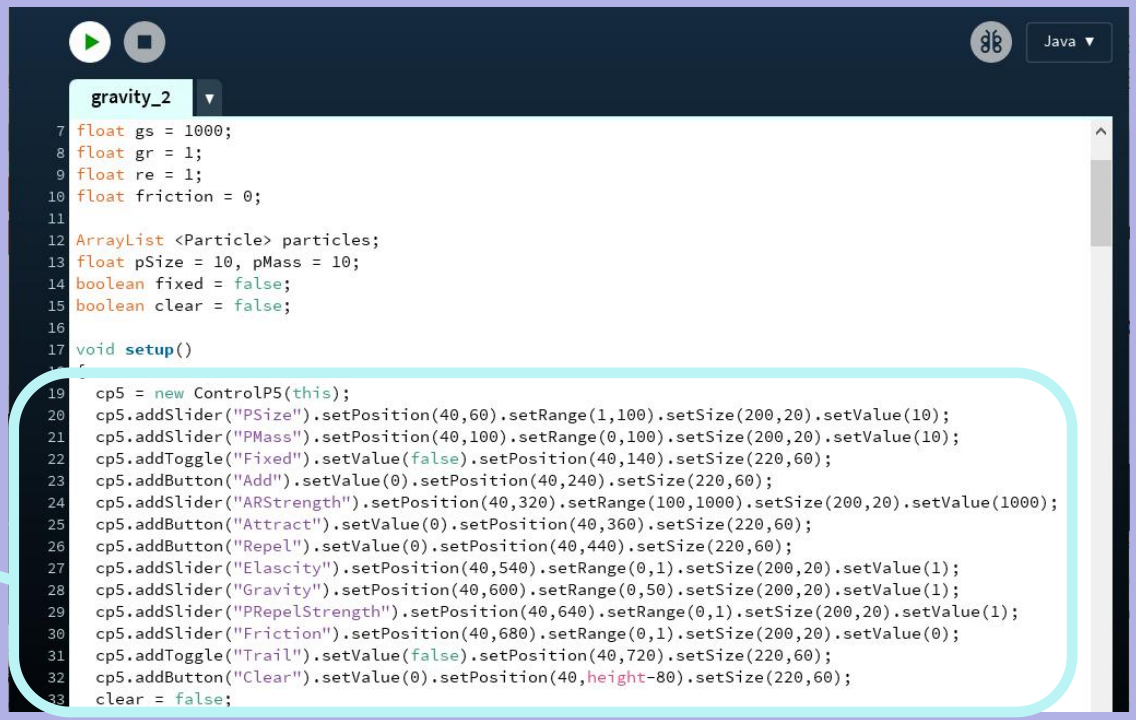
功能介绍

玩耍模式

结构设计说明

各类参数的控制基于由Andreas Schlegel所编写的controlP5库
http://www.sojamo.de/libraries/controlP5/

初始化各种slider和button.



定义粒子:

定义粒子的质量（影响粒子对其他粒子的重力）;半径; 颜色;

储存位置、速度和加速度.

储存粒子在三十帧之内的位置路径，形成好看的移动效果.
构建函数.
设置粒子属性.

每一帧粒子的显示和位置更新
（加速度决定速度 速度决定位置）
加速度在draw里更新.

```
127
128 class Particle
129 {
130   float mass = 1;
131   float rad = 10;
132   color col= color(255);
133   PVector loc;
134   PVector vel = new PVector(0,0);;
135   PVector acc = new PVector(0,0);
136   boolean fixed = false;
137   PVector[] trail = new PVector[30];
138   Particle(float x, float y)
139   {
140     loc = new PVector(x,y);
141     for (int i=0; i<30; i++) trail[i] = new PVector(x,y);
142   }
143   Particle copy() {return this;}
144   Particle setMass(float m) { mass = m; return this;}
145   Particle setRad(float r) { rad = r; return this;}
146   Particle setVel(float x, float y) { vel = new PVector(x,y); return this;}
147   Particle setAcc(float x, float y) { acc = new PVector(x,y); return this;}
148   Particle setFixed(boolean f) { fixed = f; return this;}
149   Particle setColor(color c) { col = c; return this;}
150   void next()
151   {
152     display();
153     if (!fixed)
154     {
155       if(loc.x<=300 && vel.x<0) vel.x*=-1;
156       if(loc.x>=width && vel.x>0) vel.x*=-1;
157       if(loc.y<=0 && vel.y<0) vel.y*=-1;
158       if(loc.y>=height && vel.y>0) vel.y*=-1;

160       if (trailOn) {
161         trail[0] = loc;
162         for(int i = 29; i > 0; i--) trail[i] = trail[i-1].copy();
```

```
11
12 ArrayList <Particle> particles;
13 float pSize = 10, pMass = 10;
14 boolean fixed = false;
15 boolean clear = false;
16
```

粒子信息用Arraylist储存.

```
54
55   //display
56   for (int i = 0; i < particles.size(); i++)
57   {
58     particles.get(i).next();
59   }
```

对Particles里的每个粒子调用Next(). 更新位置以及把它们画出来.

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
```

玩耍模式时用户鼠标点击所产生的吸引力、排斥力.

```
88  if (mousePressed && mouseX > 300)
89  {
90    if (tool != 0)
91    {
92      float g = gs*gs;
93      if (tool == 1) g *= -1;
94
95      PVector loc = new PVector(mouseX,mouseY);
96
97      for (int i = 0; i < particles.size(); i++)
98      {
99        PVector r = particles.get(i).loc.copy().sub(loc);
100       float d = r.mag();
101       if (d < 5 ) continue;
102       if (d < 20) d = 10;
103       float mult = g / (d*d*particles.get(i).mass);
104       particles.get(i).acc.add(r.normalize().mult(mult));
105     }
106   }
107 }
```