

# 创意编程

课后作业 (05-01) 交互设计工具

孔安梨 519430990004

# 功能介绍

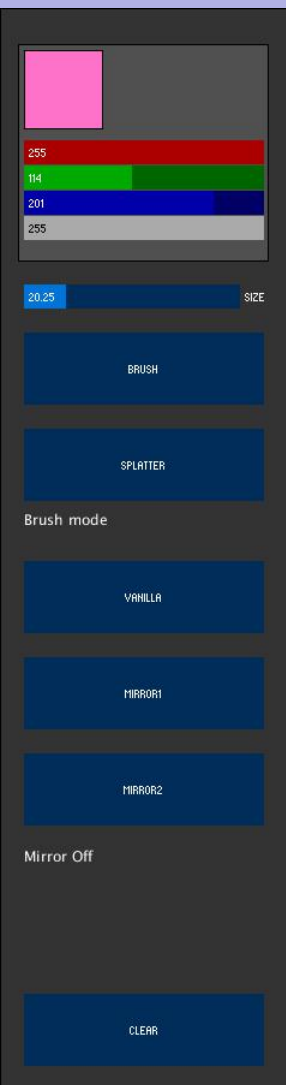
笔刷模式

可调

//颜色

//透明度

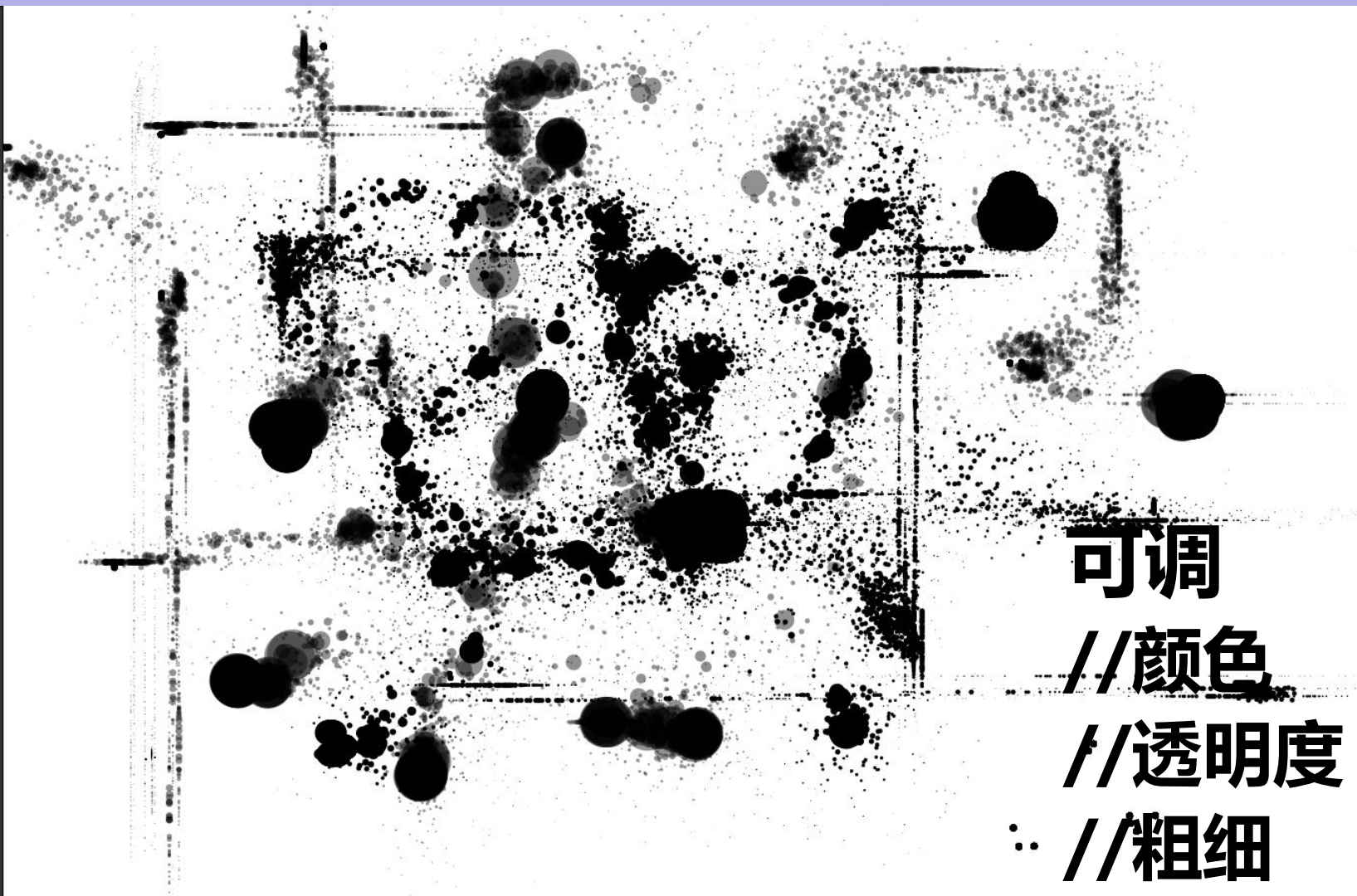
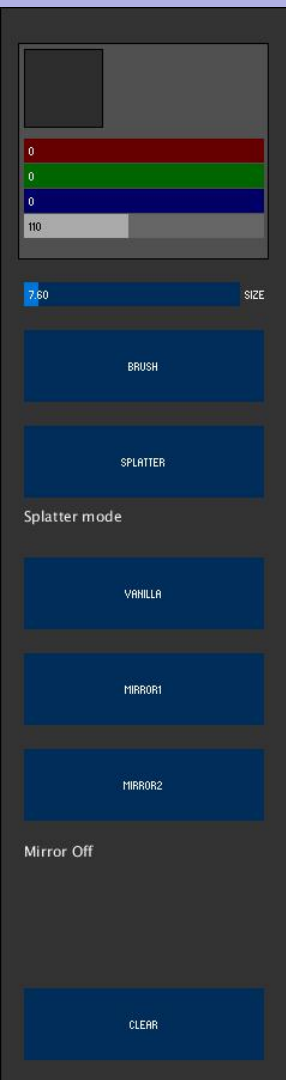
//粗细





# 功能介绍

喷溅模式



可调

//颜色

//透明度

∴ //粗细

# 功能介绍

对称模式

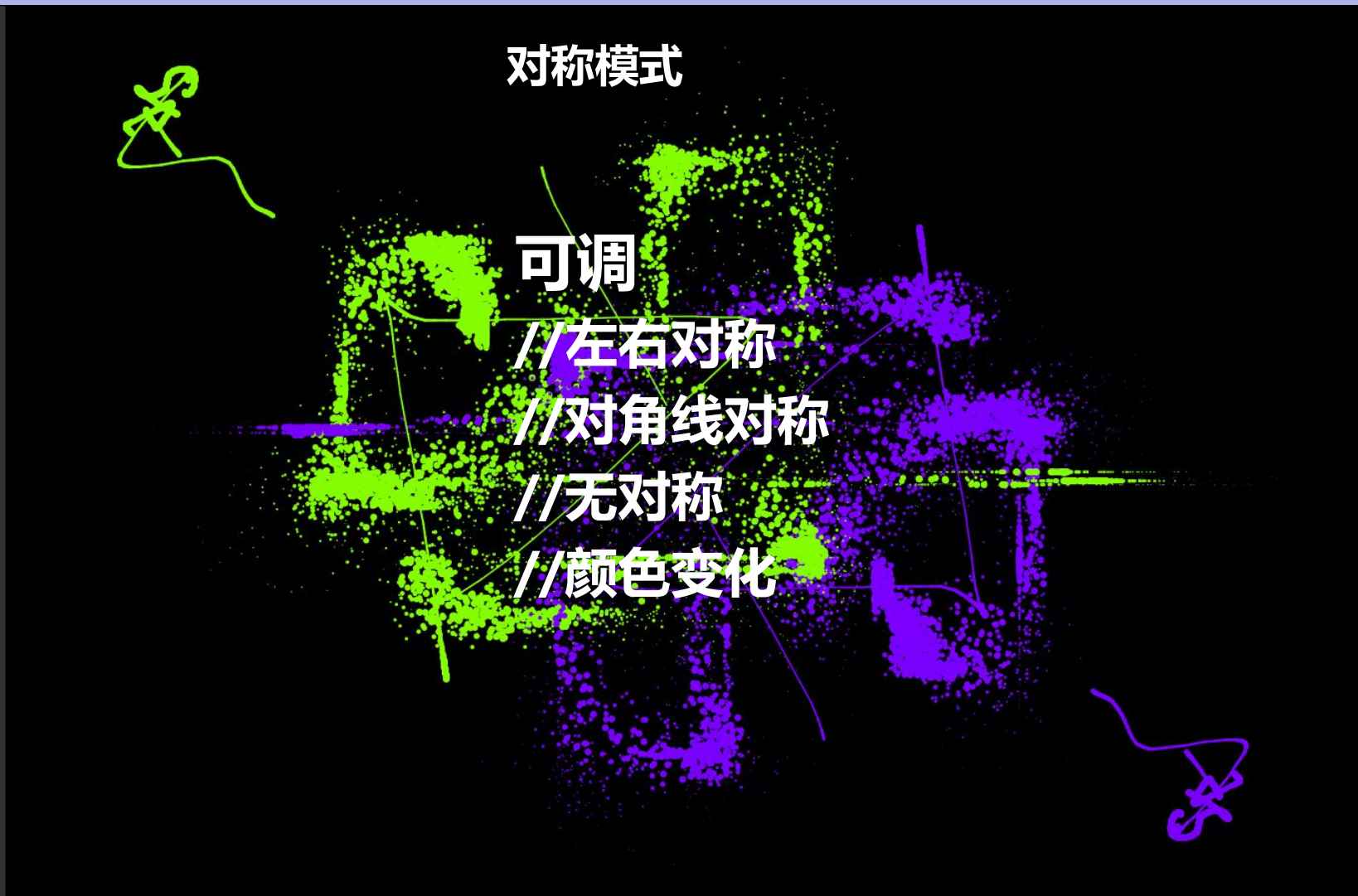
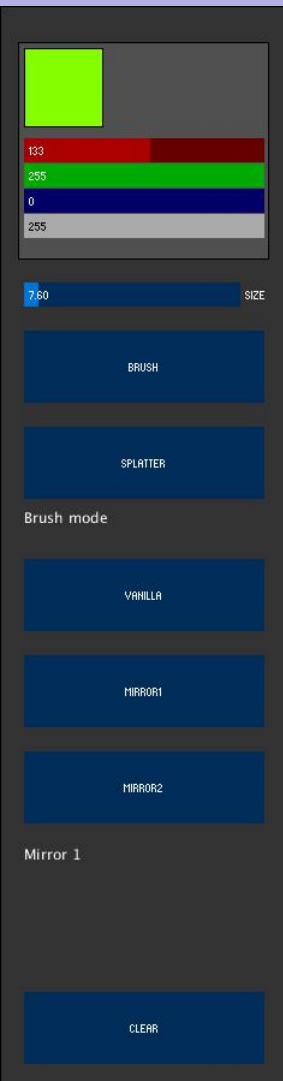
可调

//左右对称

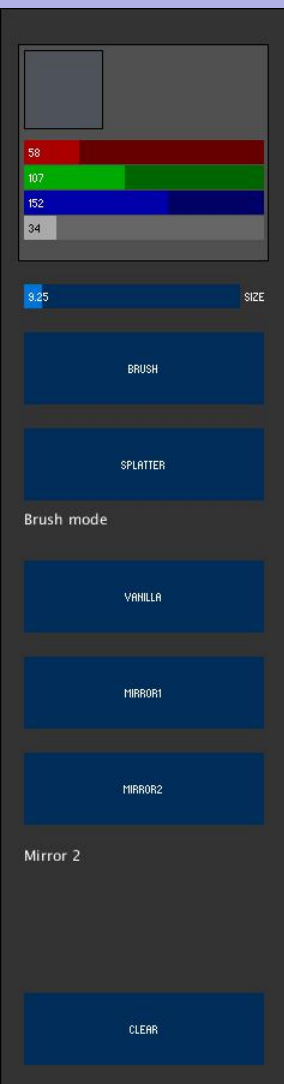
//对角线对称

//无对称

//颜色变化



# 运行实例



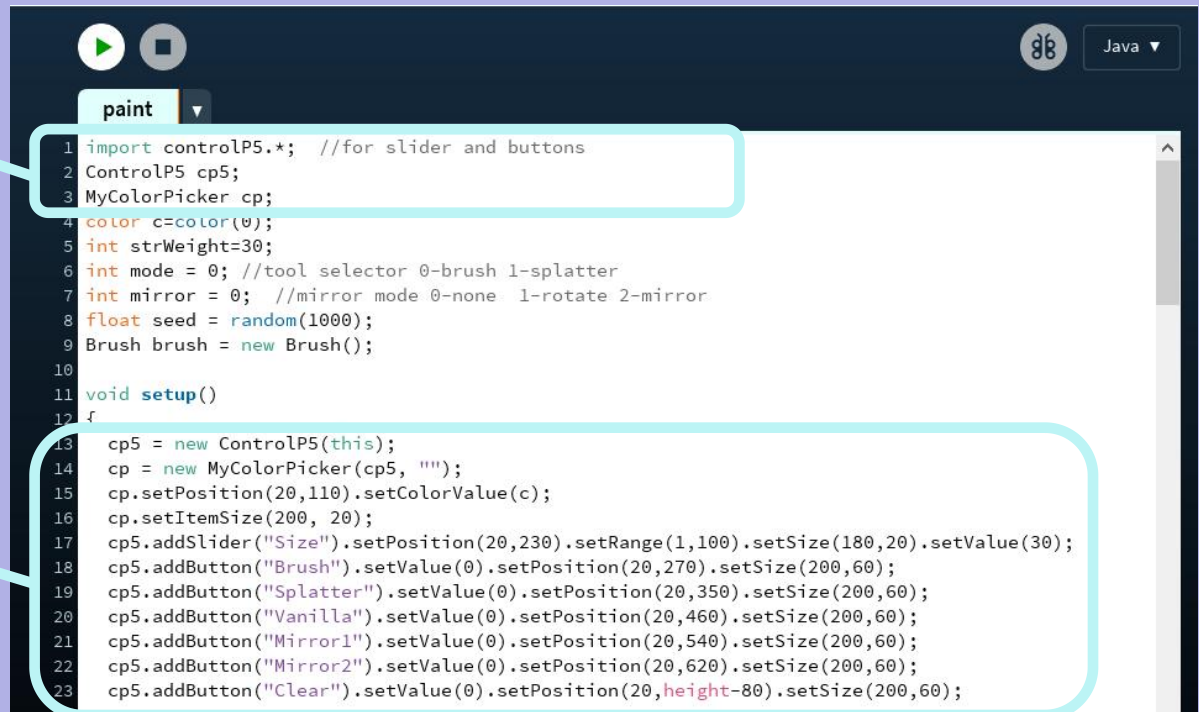


# 结构设计说明

颜色的控制(RGB模式)基于由Andreas Schlegel所编写的controlP5库  
<http://www.sojamo.de/libraries/controlP5/>

声明cp5 object, 声明cp object.  
MyColorPicker是基于controlP5  
里面colorpicker的sub-class.

初始化各种slider和button.



```
1 import controlP5.*; //for slider and buttons
2 ControlP5 cp5;
3 MyColorPicker cp;
4 color c=color(0);
5 int strWeight=30;
6 int mode = 0; //tool selector 0-brush 1-splatter
7 int mirror = 0; //mirror mode 0-none 1-rotate 2-mirror
8 float seed = random(1000);
9 Brush brush = new Brush();
10
11 void setup()
12 {
13     cp5 = new ControlP5(this);
14     cp = new MyColorPicker(cp5, "");
15     cp.setPosition(20,110).setColorValue(c);
16     cp.setItemSize(200, 20);
17     cp5.addSlider("Size").setPosition(20,230).setRange(1,100).setSize(180,20).setValue(30);
18     cp5.addButton("Brush").setValue(0).setPosition(20,270).setSize(200,60);
19     cp5.addButton("Splatter").setValue(0).setPosition(20,350).setSize(200,60);
20     cp5.addButton("Vanilla").setValue(0).setPosition(20,460).setSize(200,60);
21     cp5.addButton("Mirror1").setValue(0).setPosition(20,540).setSize(200,60);
22     cp5.addButton("Mirror2").setValue(0).setPosition(20,620).setSize(200,60);
23     cp5.addButton("Clear").setValue(0).setPosition(20,height-80).setSize(200,60);
```

增加了尺寸定义功能.

<https://forum.processing.org/one/topic/controlp5-color-picker-won-t-resize.html>

```
paint ▼
94   noStroke();
95   fill(c1);
96   circle(x,y,s);
97   seed += 0.01;
98 }
99 }
100
101 class MyColorPicker extends ColorPicker {
102   MyColorPicker(ControlP5 cp5, String theName) {
103     super(cp5, cp5.getTab("default"), theName, 0, 0, 100, 10);
104   }
105   void setItemSize(int w, int h) {
106     sliderRed.setSize(w, h);
107     sliderGreen.setSize(w, h);
108     sliderBlue.setSize(w, h);
109     sliderAlpha.setSize(w, h);
110
111     sliderGreen.setPosition(sliderGreen.getPosition()[0], sliderGreen.getPosition()[1]+h-10);
112     sliderBlue.setPosition(sliderBlue.getPosition()[0], sliderBlue.getPosition()[1]+2*(h-10));
113     sliderAlpha.setPosition(sliderAlpha.getPosition()[0], sliderAlpha.getPosition()[1]+3*(h-10));
114   }
115 }
116
```

画出页面左侧的基本板块, 显示功能状态文字.

```
paint
25 mode = 0;
26 mirror = 0;
27 size(1600, 900);
28 background(255, 245, 204);
29 }
30 void draw()
31 {
32     //panel area
33     stroke(0);
34     strokeWeight(1);
35     fill(51);
36     rect(0,0,240,height); //panel area
37     fill(80);
38     rect(15, 30, 208, 180); //slider area
39     c = cp.getColorValue();
40     fill(c);
41     square(20,35,65); //color
42     fill(255);
43     if (mode == 0) text("Brush mode",20,430);
44     else text("Splatter mode",20,430);
45     if (mirror == 0) text("Mirror Off",20,710);
46     else if (mirror == 1) text("Mirror 1",20,710);
47     else text("Mirror 2",20,710);
48 }
```



# 笔刷模式开启时:

延迟笔刷与鼠标之间的速度, 产生平滑的效果, 让速度与笔刷粗细的关系更加直接.

鼠标移动速度越快, 笔刷越细, 速度越慢, 笔刷越粗.

当镜面1功能开启时, 会同时画出x, y坐标都相反的, 颜色也相反的相同的线.

当镜面2功能开启时, 会画出面板中心左右对称的颜色相反的颜色线.

```
49 //brush effect
50 brush.x += (mouseX - brush.x)/12;
51 brush.y += (mouseY - brush.y)/12;
52 if (mousePressed && mode == 0)
53 {
54     float s = 1+2.5*strWeight/dist(brush.px, brush.py, brush.x, brush.y);
55     s=min(strWeight,s);
56     strokeWeight(s);
57     stroke(c);
58     line(brush.px, brush.py, brush.x, brush.y);
59     if (mirror == 1)
60     {
61         stroke(invertColor(c));
62         line(width-brush.px+240, height-brush.py, width-brush.x+240, height-brush.y);
63     }
64     if (mirror == 2)
65     {
66         stroke(invertColor(c));
67         line(width-brush.px+240, brush.py, width-brush.x+240, brush.y);
68     }
69 }
70 brush.px = brush.x;
71 brush.py = brush.y;
```

```
124 color invertColor(color c)
125 {
126     float r = red(c);
127     float g = green(c);
128     float b = blue(c);
129     return color(255-r,255-g,255-b);
130 }
```

# 喷溅模式开启时:

当镜面1/2功能开启时, 会同时画出与笔刷模式开启时相同的效果.

赋予鼠标周围所产生的圆的位置与圆半径在一定范围内的随机性. 鼠标滑动距离越快, 随机圆越小.

```
74 //splat effect
75 if (mousePressed && mode == 1 && mouseX > 240)
76 {
77   splatter(mouseX, mouseY, c);
78   if (mirror == 1) splatter(width-mouseX+240, height-mouseY, invertColor(c));
79   if (mirror == 2) splatter(width-mouseX+240, mouseY, invertColor(c));
80 }
81
82 void splatter(float bx, float by, color c1)
83 {
84   bx += random(-15,15);
85   by += random(-15,15);
86   float mx = 10*(abs(mouseX-pmouseX));
87   float my = 10*(abs(mouseY-pmouseY));
88   for(int i = 0; i < 80; i++){
89     seed += 0.01;
90     float x = bx+mx*(0.5-noise(seed+i));
91     float y = by+my*(0.5-noise(seed+2*i));
92     float s = 150/dist(bx, by, x, y);
93     if( s > strWeight ) s=strWeight;
94     noStroke();
95     fill(c1);
96     circle(x,y,s);
97     seed += 0.01;
98   }
99 }
```