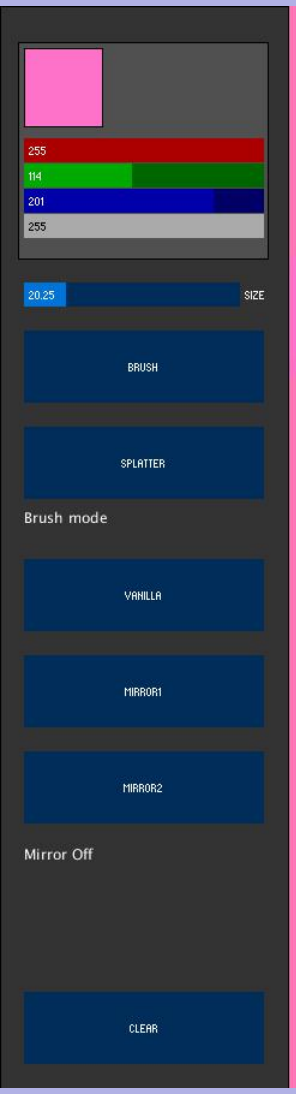

230
131
117
255
1.00
SIZE
BRUSH
SPLATTER
Splatter mode
VANILLA
MIRROR1
MIRROR2
Mirror 1
CLEAR
创意编程
课后作业 (05-01) 交互设计工具
孔安梨 5194309900004

功能介绍



笔刷模式

可调

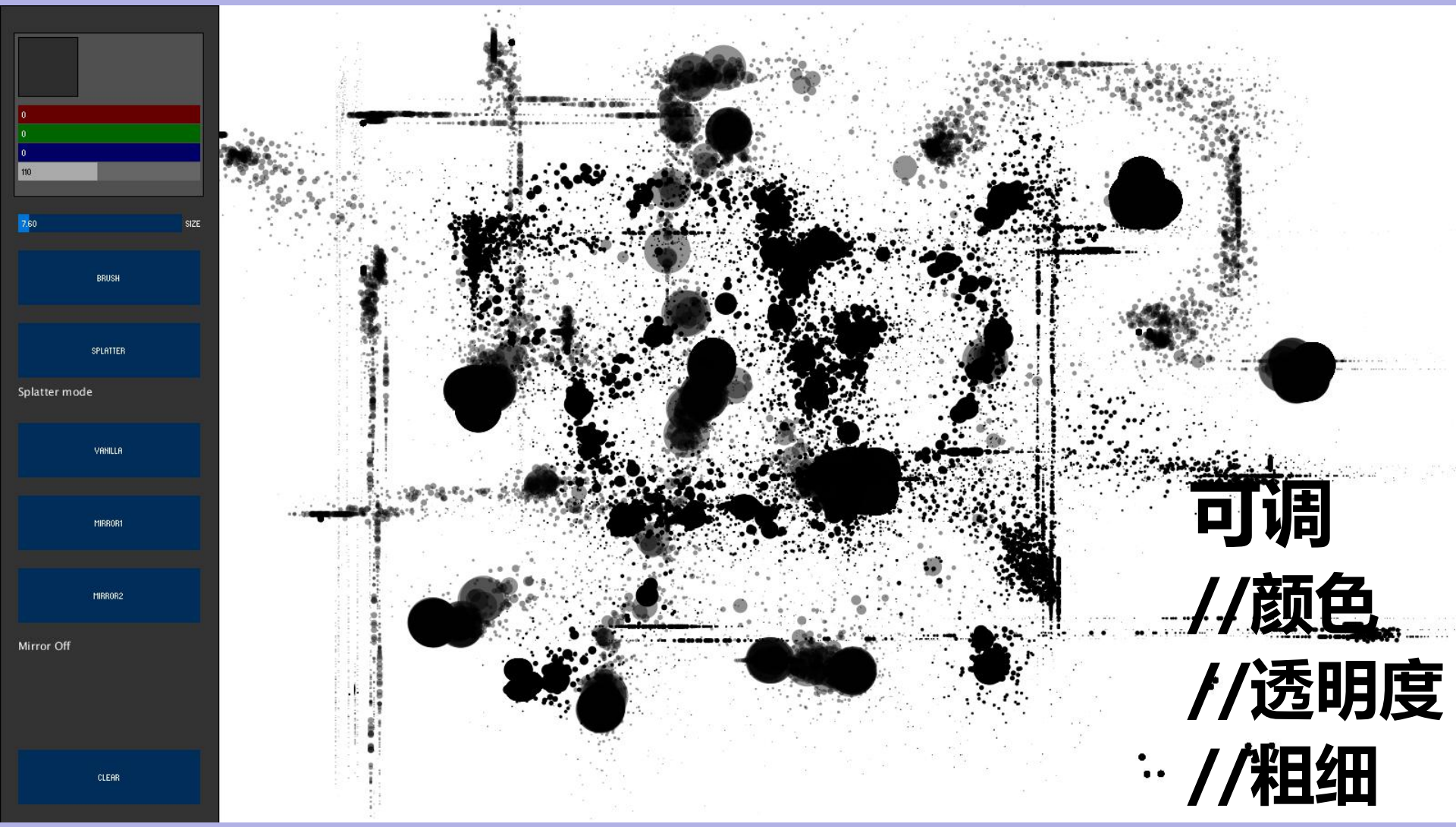
//颜色

//透明度

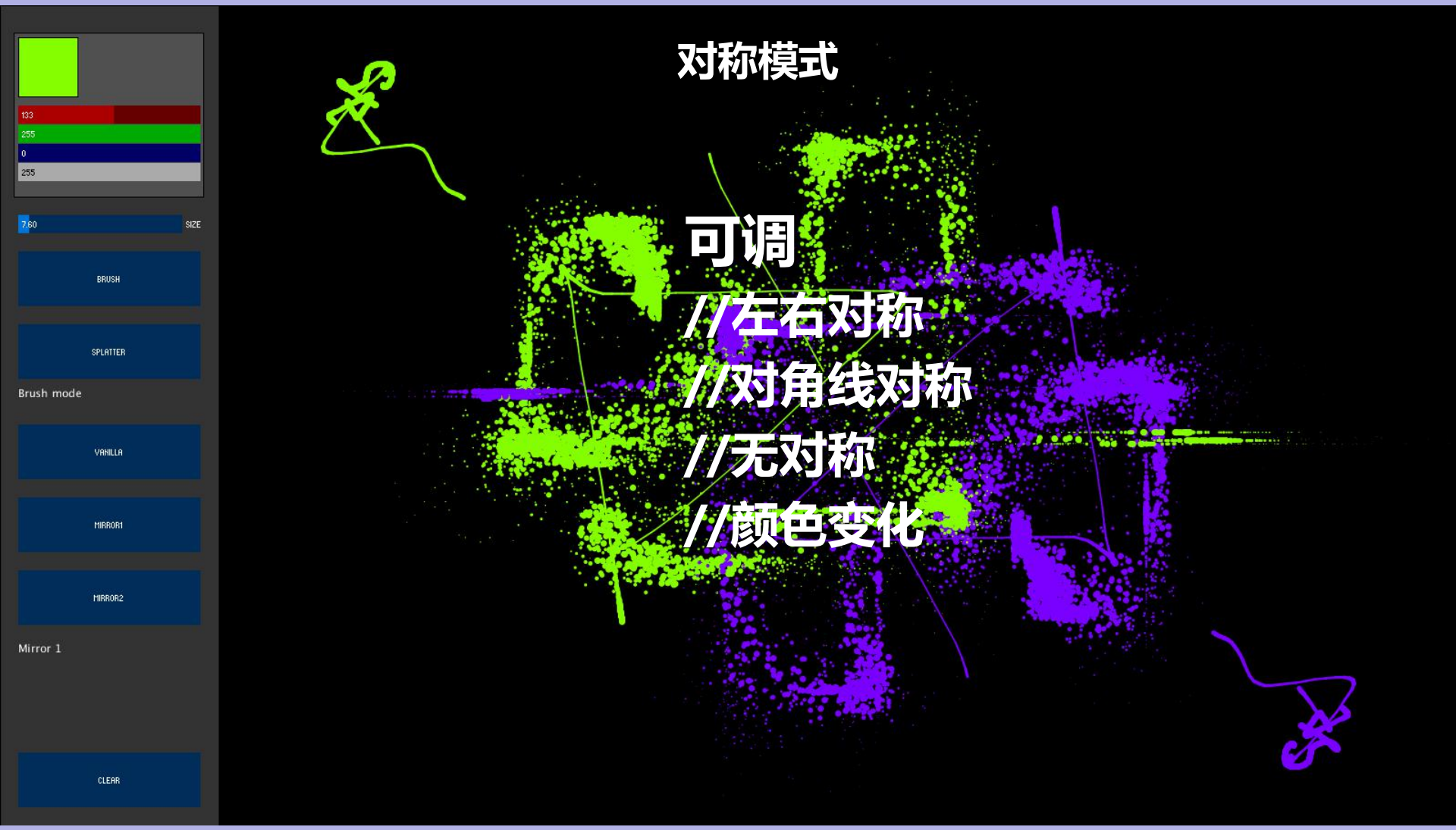
//粗细

功能介绍

喷溅模式


0
0
0
110
7.60
SIZE
BRUSH
SPLATTER
Splatter mode
VANILLA
MIRROR1
MIRROR2
Mirror Off
CLEAR
可调
//颜色
//透明度
//粗细

功能介绍


133
255
0
255
7.60
SIZE
BRUSH
SPLATTER
Brush mode
VANILLA
MIRROR1
MIRROR2
Mirror 1
CLEAR
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

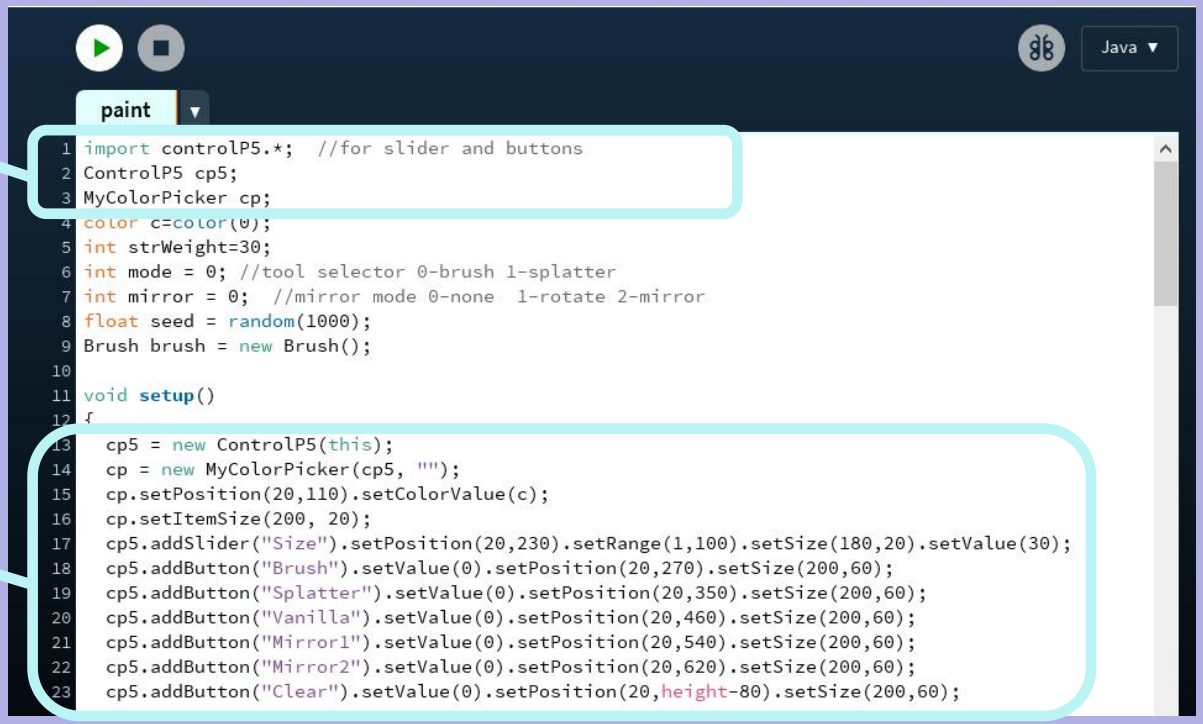

58
107
152
34
9.25
SIZE
BRUSH
SPLATTER
Brush mode
VANILLA
MIRROR1
MIRROR2
Mirror 2
CLEAR
可
//重置背景

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.



增加了尺寸定义功能.
https://forum.processing.org/one/topic/controlp5-color-picker-won-t-resize.html

paint

```
    noStroke();
    fill(c1);
    circle(x,y,s);
    seed += 0.01;
  }
}

class MyColorPicker extends ColorPicker {
  MyColorPicker(ControlP5 cp5, String theName) {
    super(cp5, cp5.getTab("default"), theName, 0, 0, 100, 10);
  }
  void setItemSize(int w, int h) {
    sliderRed.setSize(w, h);
    sliderGreen.setSize(w, h);
    sliderBlue.setSize(w, h);
    sliderAlpha.setSize(w, h);

    sliderGreen.setPosition(sliderGreen.getPosition()[0], sliderGreen.getPosition()[1]+h-10);
    sliderBlue.setPosition(sliderBlue.getPosition()[0], sliderBlue.getPosition()[1]+2*(h-10));
    sliderAlpha.setPosition(sliderAlpha.getPosition()[0], sliderAlpha.getPosition()[1]+3*(h-10));
  }
}
```

画出页面左侧的基本板块, 显示功能状态文字.

paint

```
  mode = 0;
  mirror = 0;
  size(1600, 900);
  background(255, 245, 204);
}
void draw()
{
  //panel area
  stroke(0);
  strokeWeight(1);
  fill(51);
  rect(0,0,240,height);  //panel area
  fill(80);
  rect(15, 30, 208, 180);  //slider area
  c = cp.getColorValue();
  fill(c);
  square(20,35,65);  //color
  fill(255);
  if (mode == 0) text("Brush mode",20,430);
  else text("Splatter mode",20,430);
  if (mirror == 0) text("Mirror Off",20,710);
  else if (mirror == 1) text("Mirror 1",20,710);
  else text("Mirror 2",20,710);
```

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相同的线.

```
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.

```
  if (mousePressed && mode == 1 && mouseX > 240)
  {
    splatter(mouseX, mouseY, c);
    if (mirror == 1) splatter(width-mouseX+240, height-mouseY, invertColor(c));
    if (mirror == 2) splatter(width-mouseX+240, mouseY, invertColor(c));
  }
}

void splatter(float bx, float by, color c1)
{
  bx += random(-15,15);
  by += random(-15,15);
  float mx = 10*(abs(mouseX-pmouseX));
  float my = 10*(abs(mouseY-pmouseY));
  for(int i = 0; i < 80; i++){
    seed += 0.01;
    float x = bx+mx*(0.5-noise(seed+i));
    float y = by+my*(0.5-noise(seed+2*i));
    float s = 150/dist(bx, by, x, y);
    if( s > strWeight ) s=strWeight;
    noStroke();
    fill(c1);
    circle(x,y,s);
    seed += 0.01;
  }
}
```