canvas-vue彩色丝带展示 目录

前言

- 着重点在：位置的选择和色彩的选择

推荐阅读

- 绘制随机不规则三角彩条——小谈EvanYou个人主页的实现
- Evan You

原版代码

- 暴力美学

```
<canvas></canvas>
<script>
// 事件的定义
// 如果使用触摸屏，当滚动时，将会阻止事件发生
document.addEventListener('touchmove', event => event.preventDefault());

// 获取canvas画板对象，并进行基础设置的赋值
var c = document.getElementsByTagName('canvas')[0],
    x = c.getContext('2d'),
    pr = window.devicePixelRatio || 1,
    w = window.innerWidth,
    h = window.innerHeight,
    f = 90,
    q,
    m = Math,
    r = 0,
    u = m.PI*2,
    v = m.cos,
    z = m.random

// canvas画板的长宽设置
c.width = w*pr
c.height = h*pr

// 对画板里的内容进行放大
x.scale(pr, pr)
// 图像透明度设置
x.globalAlpha = 0.6

```

```
// 清理出空间供绘图使用
function i(){
    x.clearRect(0,0,w,h)
    //三角形的起始坐标
    // f为初始值
    q=[{x:0,y:h*.7+f},{x:0,y:h*.7-f}]
    //第二个三角形的横坐标在范围内，就开始绘图
    while(q[1].x<w+f) d(q[0], q[1])
}

function d(i,j){
    // 开始绘画
    x.beginPath()
    // 连线两顶点
    x.moveTo(i.x, i.y)
    x.lineTo(j.x, j.y)

    // 求取第3个顶点
    var k = j.x + (z()*2-0.25)*f,
        n = y(j.y)
    x.lineTo(k, n)
    x.closePath()
    r-=u/-50
    x.fillStyle = '#'+(v(r)*127+128<<16 | v(r+u/3)*127+128<<8 |
v(r+u/3*2)*127+128).toString(16)
    x.fill()
    q[0] = q[1]
    q[1] = {x:k,y:n}
}

function y(p){
    var t = p + (z()*2-1.1)*f
    return (t>h||t<0) ? y(p) : t
}

document.onclick = i
document.ontouchstart = i
i()

</script>
```

自用版本

- 符合个人使用习惯

```
<canvas></canvas>
<script>
document.addEventListener('touchmove', event => event.preventDefault());

let ctx = document.getElementsByTagName('canvas')[0],
    c = ctx.getContext('2d'),
    basePixel = window.devicePixelRatio || 1,
    width = window.innerWidth,
    height = window.innerHeight,
    startOffset = 90,
    coordinateArr,
```

```
    startRadius = 0,
    radius = Math.PI * 2;

ctx.width = width * basePixel,
ctx.height = height * basePixel;

c.scale(basePixel, basePixel),
c.globalAlpha = 0.6;

// 事件调用
// 点击重绘和拖动重绘
document.onclick = initialPotray;
document.ontouchstart = initialPotray;
// 如果没有事件发生，初始绘制
initialPotray();

function initialPotray(){
    c.clearRect(0,0,width,height);
    // first
    // 基于0.7height的位置对称
    coordinateArr = [
        {x:0, y: height * .7 + startOffset},
        {x:0, y: height * .7 - startOffset}
    ];
    // 让最后一个点跑出去，不至于最后狭小
    while(coordinateArr[1].x < width + startOffset)
drawTraingleImage(coordinateArr[0],coordinateArr[1]);
}

function drawTraingleImage(coor1,coor2){
    c.beginPath();
    c.moveTo(coor1.x, coor1.y);
    c.lineTo(coor2.x, coor2.y);
    // x取值在x2+[-22.5,157.5]，如果大于宽度，跳出循环
    // y在y2+[-22.5,157.5]之间，如果大于高度，重新抽取
    let coor3 = {
        x: coor2.x + (Math.random()*2 - 0.25) * startOffset,
        y: reCalculate(coor2.y)
    };

    c.lineTo(coor3.x, coor3.y);
    c.closePath();

    startRadius -= radius / -50;
    // RGB转换
    // 内容数字可改
    c.fillStyle = '#' + (
        Math.cos(startRadius)*127 + 128<<16 |
        Math.cos(startRadius + radius/3)*127 + 128<<8 |
        Math.cos(startRadius + radius/3 * 2)*127 + 128
    ).toString(16);

    c.fill();

    // 更换坐标
    coordinateArr[0] = coordinateArr[1],
    coordinateArr[1] = coor3;
}
```

```
69
70  function reCalculate(yNum){
71      let tmp = yNum + (Math.random()*2 - 1.1) * startOffset;
72      return (tmp > height || tmp < 0) ? reCalculate(yNum) : tmp;
73  }
74
75  </script>
```

结果展示