**计算机图形学**

**实验报告**

课程名称： 计算机图形学实验报告

专 业： 计算机科学与技术 班级： 2018-5

姓 名： 刘澳

学 号： 201801011312

**山 东 科 技 大 学**

**2021年 6月 27日**

# 实验五 直线段裁剪

1. 实验目的
2. 掌握直线段端点编码方法。
3. 掌握Cohen-Sutherland直线段裁剪算法。
4. 实验要求
5. 定义二维坐标系原点位于屏幕中心，x轴水平向右为正，y轴铅直向上为正。
6. 在客户区中央固定绘制颜色为RGB(128,0,0)的3像素宽的矩形代表裁剪窗口。裁剪窗口的左上角点为(-300,100) ,右下角点为(300,-100)。
7. 使用鼠标在屏幕上动态绘制任意直线段。
8. 选择裁剪按钮根据直线段和窗口的相对位置，对直线段进行裁剪，得到位于窗口内的直线段，删除窗口外的直线段。
9. 直线段绘制之前，裁剪按钮无效;直线段绘制之后，裁剪按钮有效。
10. 关键程序代码

<script *type*="text/javascript">

    var canvas = document.**getElementById**("myCanvas");

    var ctx = canvas.**getContext**('2d');

    ctx.fillStyle = "rgba(232,221,203,0.5)";

    ctx.**fillRect**(0, 0, 1000, 1000);

    ctx.**rect**(300, 200, 300, 200);

    ctx.**clip**();

    position = [];

    canvas.**addEventListener**('click', **click**);

*//修正函数*

**HTMLCanvasElement**.prototype.**getMousePosition** = function (event) {

        var totalOffsetX = 0;

        var totalOffsetY = 0;

        var canvasX = 0;

        var canvasY = 0;

        var currentElement = this;

        do {

            totalOffsetX += currentElement.offsetLeft;

            totalOffsetY += currentElement.offsetTop;

            currentElement = currentElement.offsetParent;

        } while (currentElement);

        canvasX = event.pageX - totalOffsetX;

        canvasY = event.pageY - totalOffsetY;

*// Fix for variable canvas width*

        canvasX = **Math**.**round**(canvasX \* (this.width / this.offsetWidth));

        canvasY = **Math**.**round**(canvasY \* (this.height / this.offsetHeight));

        return { x: canvasX, y: canvasY };

    };

    function **click**(e) {

        position.**push**(canvas.**getMousePosition**(e));

        if (position.length == 2) {

            console.**log**("111");

**drawLine**(position[0].x, position[1].x, position[0].y, position[1].y);

            position = [];

        }

    }

    function **drawLine**(x1, x2, y1, y2) {

        ctx.strokeStyle = "black";

        ctx.**moveTo**(x1, y1);

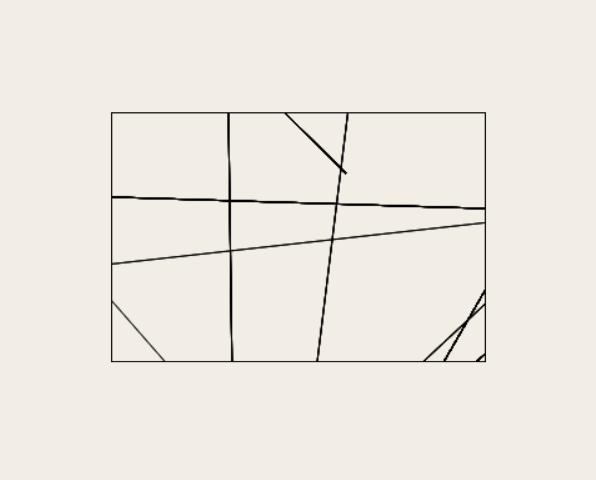
        ctx.**lineTo**(x2, y2);

        ctx.**stroke**();

    }

</script>

1. 程序运行结果、界面等



1. 思考与总结等

在编写Cohen-Sutherland算法时没有遇到什么问题，无非是先编码，后位运算判断位置，复杂计算求交。在这个算法里面最应该学习的思想是编码思想。对于这个编码思想在计算机图形学中有大量的应用，用编码来表示图形位置或图形状态，将图形编码化，可视化的东西计算化，这是实现计算机图形学的基本思想之一。