**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**

# 实验三 交互式绘制多边形

1. 实验目的
2. 掌握双缓冲绘图技术。
3. 掌握人机交互技术。
4. 掌握填充动态多边形的有效边表算法。
5. 实验要求
6. 使用鼠标在屏幕客户区绘制任意点数的多边形。要求使用橡皮筋技术动态绘制每条边;鼠标移动过程中按下Shift键时可绘制垂直边或水平边;将多边形的终点移动到多边形的起点时自动封闭多边形;在绘制多边形的过程中，状态栏动态显示鼠标光标的位置坐标。。
7. 当开始绘制多边形时，更改鼠标光标为十字光标，多边形绘制完毕后恢复为箭头光标。
8. 多边形闭合后自动调用有效边表算法填充多边形内部区域。
9. 实验步骤

先在画布上取一个点并记录下来，然后连续在画布上取点直至图形封闭。

封闭后再重新绘制这些点和线，并且填充起来。

1. 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);

    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) {

        var pos = canvas.**getMousePosition**(e);

        if (position.length == 0)

            position.**push**(pos);

        else if (**isClose**(position, pos))

        {

            console.**log**("11111");

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

**draw**(position);

            position = [];

        }

        else

            position.**push**(pos);

        if (position.length >= 2) {

**drawLine**(position[position.length-2].x, position[position.length-1].x, position[position.length-2].y, position[position.length-1].y);

        }

*// arraySort(position);*

    }

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

        ctx.**moveTo**(x1, y1);

        ctx.**lineTo**(x2, y2);

        ctx.**stroke**();

    }

    function **draw**(position)

    {

        ctx.**beginPath**();

        for(var i = 0; i < position.length; i++)

        {

            ctx.**lineTo**(position[i].x, position[i].y);

        }

        ctx.**closePath**();

        ctx.fillStyle = "rgba(254, 67, 101, 0.8)"; *//填充封闭图形的颜色,*

        ctx.**fill**(); *//填充图形 ( 后绘制的图形会覆盖前面的图形, 绘制时注意先后顺序 )*

    }

    function **isClose**(arr, pos) {

        if ( **Math**.**abs**(arr[0].x-pos.x)<3 && **Math**.**abs**(arr[0].y-pos.y)<3)

        {

            console.**log**("0000");

            return true;

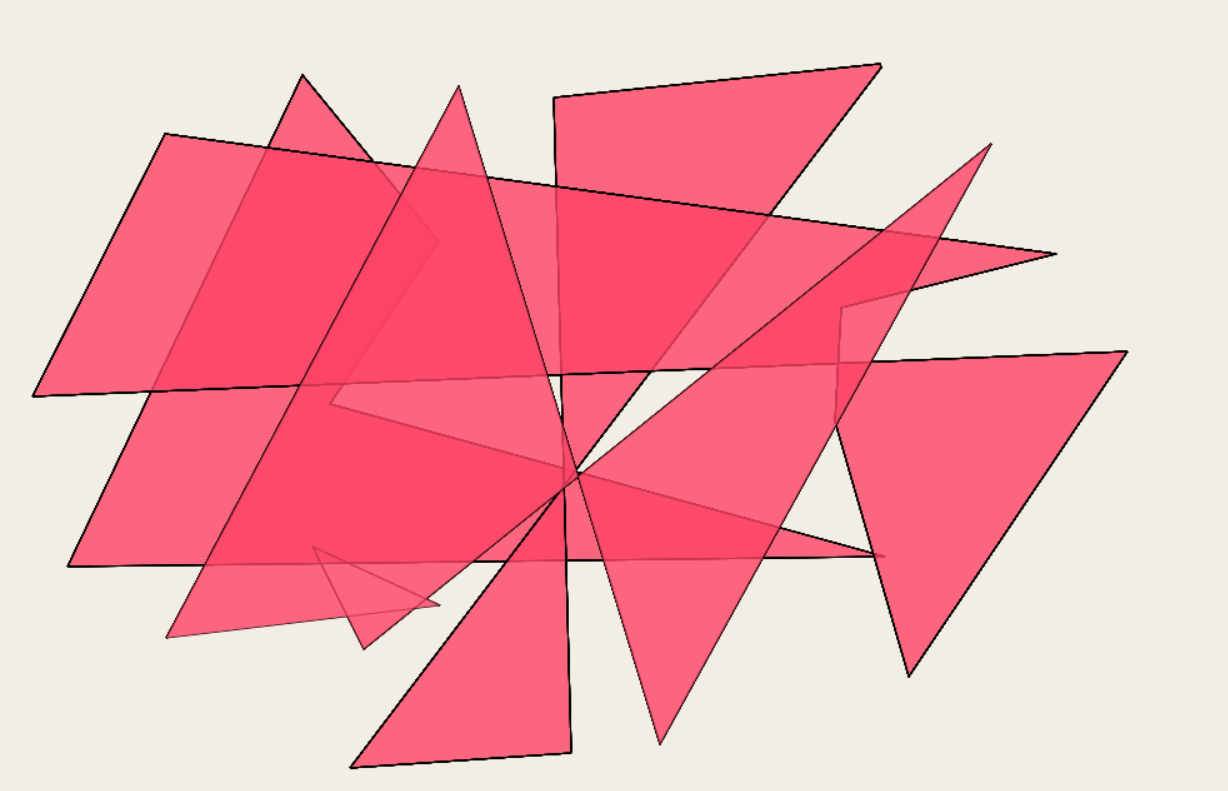
        }

        return false;

    }

</script>

1. 程序运行结果、界面等



1. 思考与总结等

这次实验是绘制多边形，期间遇到了绘制最后一根线时总是不能进行填充。后来采用了另一种方法，再绘制最后一根线时重新绘制，然后再进行填充。这样就能正确的将封闭图形填充起来。