**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**

# 实验二 绘制任意斜率的直线段

1. 实验目的
2. 掌握二维坐标系模式映射方法。
3. 掌握任意斜率直线段的中点Bresenham扫描转换算法。
4. 掌握CLine直线类的设计方法。
5. 掌握CLine直线类的设计方法。
6. 实验要求
7. 设计CLine直线类，其数据成员为直线段的起点坐标P。和终点坐标P，成员函数为MoveTo()和LineTo()函数。
8. CLine类的Line To()函数使用中点Bresenham算法绘制任意斜率k的直线段，包括k=士∞、k>1、0≤k≤1、-1≤k<0和k< -1五种情况。
9. 在屏幕客户区按下鼠标左键选择直线的起点，保持鼠标左键按下并移动鼠标到另一位置，弹起鼠标左键绘制任意斜率的直线段。
10. 在状态栏动态显示鼠标光标移动时的位置坐标。
11. 实验步骤

先在画布上取一个点并记录下来，第二次取点的时候用Breshman算法将重点起点之间的所有点迭代并绘制出来。

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)

    {

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

        if (position.length == 2)

        {

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

            position = [];

        }

    }

    function **drawClickLine**(x1, x2, y1, y2)

    {

*// 使用 Bresenham 算法画任意斜率的直线（包括起始点，不包括终止点）*

        var x = x1;

        var y = y1;

        var dx = **Math**.**abs**(x2 - x1);

        var dy = **Math**.**abs**(y2 - y1);

        var s1 = x2 > x1 ? 1 : -1;

        var s2 = y2 > y1 ? 1 : -1;

        var interchange = false;    *// 默认不互换 dx、dy*

        if (dy > dx)                *// 当斜率大于 1 时，dx、dy 互换*

        {

            var temp = dx;

            dx = dy;

            dy = temp;

            interchange = true;

        }

        var p = 2 \* dy - dx;

        for(var i = 0; i < dx; i++)

        {

**putpixel**(x, y);

            if (p >= 0)

            {

                if (!interchange)        *// 当斜率 < 1 时，选取上下象素点*

                    y += s2;

                else                    *// 当斜率 > 1 时，选取左右象素点*

                    x += s1;

                p = p + 2 \* dy - 2 \* dx;

            }

            if (!interchange)

                x += s1;                *// 当斜率 < 1 时，选取 x 为步长*

            else

                y += s2;                *// 当斜率 > 1 时，选取 y 为步长*

            p += 2 \* dy;

        }

    }

    function **putpixel**(x, y)

    {

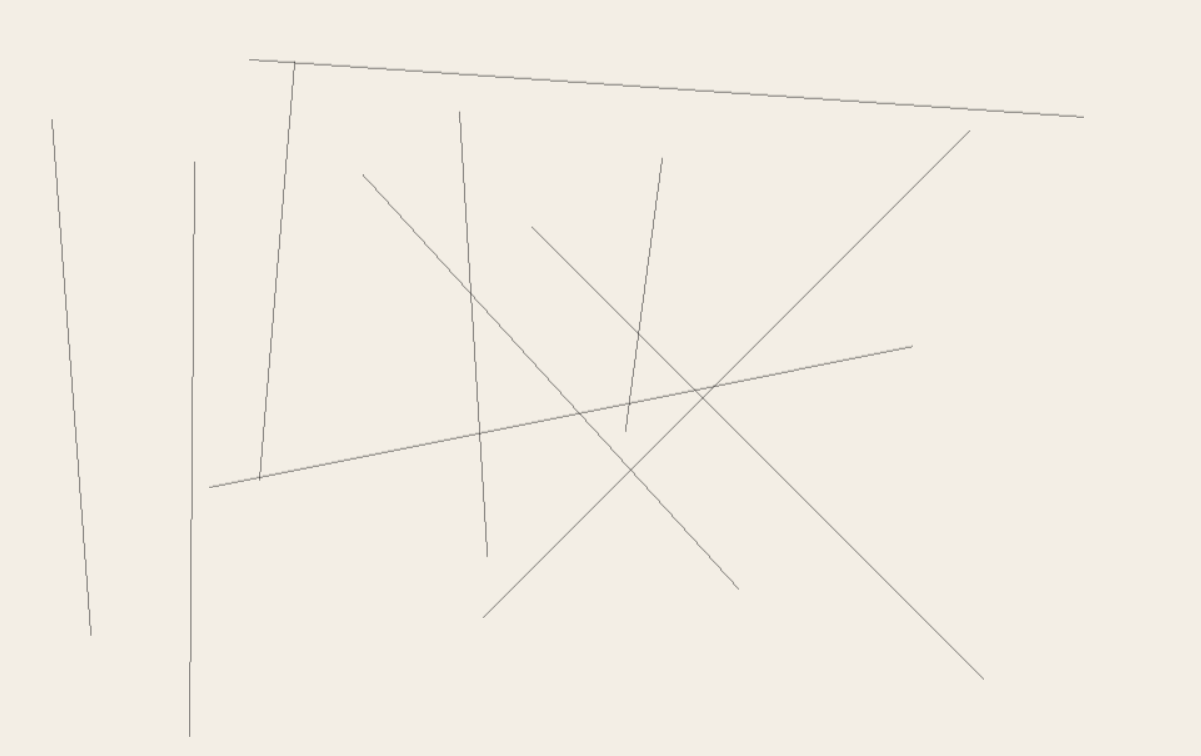
        ctx.fillStyle = "black";

        ctx.**fillRect**(x, y, 1, 0.5);

    }

</script>

1. 程序运行结果、界面等



1. 思考与总结等

上课的时候没好好听，对此算法并不是很熟悉，通过实验课重新学了一遍并且实践出来，这样加深了我对Breshman算法的理解。

自此我写出这次实验之后，深深的意识到了上课听讲的重要性，以后一定好好听课，这样在实践、应用的时候才有理论基础，而不是对于别人成果的简单的重复！