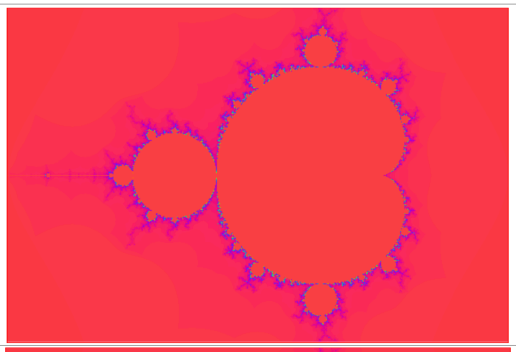
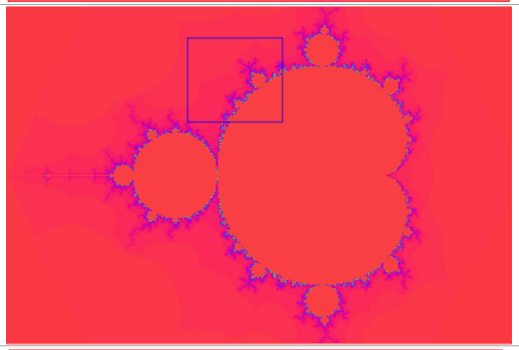
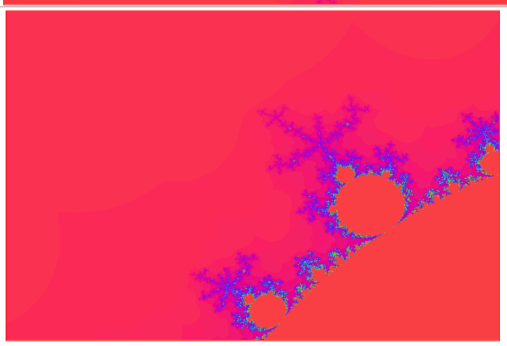
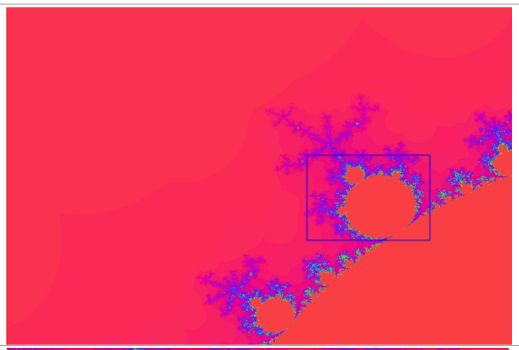
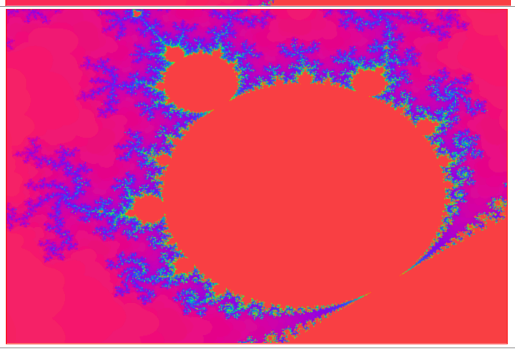
資管三 陳俊元 S0761127

HW2

初始顯示的畫面	
用滑鼠選取範圍	
將選取的範圍放大	
再用滑鼠選取範圍	
範圍放大	

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HW2</title>
</head>
<body>
    <div id="info">info</div>
    <canvas id="fractal" width="600" height="400" style="border: red solid
1px" />
    <script>
        var canvas = document.querySelector("#fractal");
        var ctx = canvas.getContext("2d");
        var info = document.querySelector("#info");
        var mouseX = 0;
        var mouseY = 0;
        info.innerHTML= `<h3> ${mouseX}, ${mouseY} </h3>`;
        var img = ctx.getImageData(0, 0, canvas.clientWidth,
canvas.clientHeight);
        var pixels = img.data;
        var fractal_viewport =
        {
                xmin: -2,
                xmax: 1,
                ymin: -1,
                ymax: 1
        };
        var  maxIterations = 256;
        var fw = 3;
        var fh = 2;
        var rubberband = false;
        var band =
        {
            x:0,
            y:0,
            width:0,
            height:0
        }
        let colorA= {r: 0.5,  g:0.5, b:0.5 };
        let colorB= {r: 0.5,  g:0.5, b:0.5 };
        let colorC= {r: 1.0,  g:1.0, b:1.0 };
        let colorD= {r: 0.0,  g:0.33, b:0.67 };

        animate();
        canvas.addEventListener("mousedown",onMouseDown);
        canvas.addEventListener("mousemove",onMouseMove);
        canvas.addEventListener("mouseup",onMouseUp);

        function animate()
        {
            ctx.clearRect(0,0,canvas.clientWidth,canvas.clientHeight);
            fractal();
            if(rubberband)
            {
                ctx.strokeStyle="blue"
                ctx.strokeRect(band.x, band.y, band.width, band.height);
            }
            requestAnimationFrame(animate);
        }

        function fractal(){
            for(let y = 0; y < canvas.clientHeight; y++)
            {
                for(let x = 0; x < canvas.clientWidth; x++)
                {
                    //z = z ^ 2 + c
                    let c = pixelToComplexPoint(x, y);
                    z =
                    {
                        x:0,
                        y:0,
                    }
                    for(var i = 0; i <  maxIterations ; i++)
                    {
                        if(z.x + z.x + z.y * z.y > 4)
                            break;
                        let temp = z.x * z.x - z.y * z.y + c.x;
                        z.y = 2*z.x * z.y + c.y;
                        z.x = temp;
                    }
                    /*var color=
                    {
                        r : Math.floor((i/maxIterations) * 256),
                        g : 255,
                        b : 255,
                        a : 255
                    }*/
                    let color = palette(i/maxIterations, colorA, colorB,
colorC, colorD);
                    writePixel(x,y,color.r, color.g, color.b);
                }
            ctx.putImageData(img, 0, 0);

            }
        }



        function writePixel(x,y,r,g,b, a)
        {
            let index = ( y * canvas.clientWidth + x) *4;
            pixels[index] = r;
            pixels[index+1] = g;
            pixels[index+2] = b;
            pixels[index+3] = 255;

        }

        function pixelToComplexPoint(x, y)
        {
            let c =
            {
                x: x / canvas.clientWidth * fw + fractal_viewport.xmin,
                y: fractal_viewport.ymin + y / canvas.clientHeight * fh
            };
            return c;
        }


        function onMouseUp(event)
        {
            let z1 = pixelToComplexPoint(band.x, band.y);
            let z2 = pixelToComplexPoint(band.x+band.width,
band.y+band.width*0.75);

            fractal_viewport.xmin = z1.x;
            fractal_viewport.ymin = z1.y;
            fractal_viewport.xmax = z2.x;
            fractal_viewport.ymax = z2.y;
            fw = Math.abs(z2.x - z1.x);
            fh = Math.abs(z2.y - z1.y);
            rubberband = false;
            canvas.removeEventListener("mouseMove", onMouseMove);
            canvas.removeEventListener("mouseup", onMouseUp);
        }

        function onMouseDown(event)
        {
            rubberband = true;
            mouseX = event.clientX - canvas.offsetLeft;
            mouseY = event.clientY - canvas.offsetTop;
            info.innerHTML= `<h3> ${mouseX}, ${mouseY} </h3>`;
            band.x = mouseX;
            band.y = mouseY;
            canvas.addEventListener("mouseMove", onMouseMove);
            canvas.addEventListener("mouseup", onMouseUp);
        }

        function onMouseMove(event)
        {
            mouseX = event.clientX - canvas.offsetLeft;
            mouseY = event.clientY - canvas.offsetTop;
            info.innerHTML= `<h3> ${mouseX}, ${mouseY} </h3>`;
            if(rubberband)
            {
                band.width = mouseX - band.x;
                band.height = mouseY - band.y;
            }
        }




        function palette(t, a, b, c, d )
        {
            let color =
            {
                r: Math.floor((a.r +b.r * Math.cos(6.28318 * (c.r
*t+d.r)))*255),
                g: Math.floor((a.g +b.g * Math.cos(6.28318 * (c.g
*t+d.g)))*255),
                b: Math.floor((a.b +b.b * Math.cos(6.28318 * (c.b
*t+d.b)))*255)
            }
            return color;
        }

    </script>
</body>
</html>
```