```
"use strict";
var canvas;
var gl;
var positions;
var numTimesToSubdivide=0;
var bufferId;

init();

function init()
{
    var canvas = document.getElementById(
"gl-canvas" );

    gl = canvas.getContext('webgl2');
    if ( !gl ) { alert( "WebGL isn't
available" ); }

    //
    //  Configure WebGL
    //
    gl.viewport( 0, 0, canvas.width,
canvas.height );
    gl.clearColor( 1.0, 1.0, 1.0, 1.0 );

    //  Load shaders and initialize
attribute buffers
```

```
    var program = initShaders( gl,
"vertex-shader", "fragment-shader" );
    gl.useProgram( program );

    // Load the data into the GPU

    var bufferId = gl.createBuffer();
    gl.bindBuffer( gl.ARRAY_BUFFER,
bufferId );
    //gl.bufferData( gl.ARRAY_BUFFER,
flatten(positions), gl.STATIC_DRAW );
    gl.bufferData( gl.ARRAY_BUFFER ,
8*Math.pow(3, 6) , gl.STATIC_DRAW );

    // Associate out shader variables with
our data buffer

    var positionLoc =
gl.getAttribLocation( program, "aPosition"
);
    gl.vertexAttribPointer( positionLoc,
2, gl.FLOAT, false, 0, 0 );
    gl.enableVertexAttribArray(
positionLoc );


document.getElementById("slider").onchange
```

```
= function(event) {
        numTimesToSubdivide =
parseInt(event.target.value);
        render();
    };

    render();
};

function divFlake(left, right, count)
{
    var sqrt3d2 = 0.87;
    var pos1 = mix(left,right,0.33);
    var pos2 = mix(left,right,0.67);

    if (count==0) {
       positions.push(left);
       positions.push(pos1);
       positions.push(pos1);
       positions.push(pos2);
       positions.push(pos2);
       positions.push(right);
    } else {
       // calc top and make sure it is a
vec2
       var len = pos2[0] - pos1[0];
       var top = vec2( pos1[0]+len/2 ,
len*sqrt3d2 );
```

```
            positions.push(pos1);
            positions.push(top );
            positions.push(top );
            positions.push(pos2);
            --count;
            // divide the left and right
            divFlake(left,pos1,count);
            divFlake(pos2,right,count);
        }
    }

    function render() {

        //var vertices=[
        //vec2( -1.00 ,  0.00 ),
        //vec2( -0.33 ,  0.00 ),
        //vec2( -0.33 ,  0.00 ),
        //vec2(  0.00 ,  0.00 ),
        //vec2(  0.00 ,  0.00 ),
        //vec2(  0.33 ,  0.00 ),
        //vec2(  0.33 ,  0.00 ),
        //vec2(  1.00 ,  0.00 )
        //];

        positions = [];
        //positions.push( vertices[0] );
        //positions.push( vertices[1] );
        //positions.push( vertices[2] );
    //positions.push( vertices[3] );
    //positions.push( vertices[4] );
    //positions.push( vertices[5] );
    //positions.push( vertices[6] );
    //positions.push( vertices[7] );

    var left  = vec2( -1.0 ,  0.0);
    var right = vec2(  1.0 ,  0.0);

divFlake(left,right,numTimesToSubdivide);
    //divFlake(left,right,1);

    gl.bufferSubData(gl.ARRAY_BUFFER, 0,
flatten(positions));
    gl.clear( gl.COLOR_BUFFER_BIT );
    gl.drawArrays( gl.LINES, 0,
positions.length );
    positions = [];
}
```