

Universidad Rey Juan Carlos

Gráficas y Visualización 3D

Práctica 2: *Draw a Colored Point with a Mouse Click*

Katia Leal Algara

INTRODUCCIÓN:

Partiendo del código “click_points.html” realiza las modificaciones necesarias para dibujar puntos en las coordenadas indicadas por la posición del puntero del ratón al hacer click. Debes llamar a la página “02_click_colored_points.html”. En este caso, en lugar de usar el buffer de datos, a los shaders se les pasa la información por cada vértice dibujado.

1. Declaración del fragment shader.

```
<script id="shader-fs" type="x-shader/x-fragment">
    precision mediump float;
    uniform vec4 uFragColor;
    void main(void) {
        gl_FragColor = uFragColor;
    }
</script>
```

2. La función *initWebGL()*

```
function initWebGL() {
    canvas = document.getElementById("my-canvas");
    try {
        gl = canvas.getContext("webgl") ||
            canvas.getContext("experimental-webgl");
    } catch (e) {
    }
    if (gl) {
        setupWebGL(); // No cambia
        initShaders(); // No cambia
        registerEventHandler();
    } else {
        alert("Error: Your browser does not appear to support WebGL.");
    }
}
```

3. La función *registerEventHandler()*

```
function registerEventHandler() {
    // Aquí faltan un par de líneas de código
    // Register function (event handler) to be called on a mouse press
    canvas.onmousedown=function(ev){click(ev,gl,canvas,vertexPositionAttribute,
    fragColorAttribute)};
}
```

4. La función *click()*

```
function click(ev, gl, canvas, vertexPositionAttribute) {
    // Aquí el código para almacenar las coordenadas
    ...
    mouseClickPoints.push([x, y, z]);

    // Store the color
    if(x >= 0.0 && y >= 0.0) { // First quadrant
        mouseClickColors.push([1.0, 1.0, 0.0, 1.0]); // Yellow
    } else if(x < 0.0 && y < 0.0) { // Third quadrant
        mouseClickColors.push([0.0, 1.0, 0.0, 1.0]); // Green
    } else {
        // Others
        mouseClickColors.push([1.0, 0.0, 1.0, 1.0]); // Purple
    }

    // Clear <canvas>
    gl.clear(gl.COLOR_BUFFER_BIT);
    var len = mouseClickPoints.length;
    for(var i = 0; i < len; i++) {
        var xy = mouseClickPoints[i];
        var rgba = mouseClickColors[i];

        // Pass the position of a point to aPosition variable
        ...
        // Pass the color of a point to uFragColor variable
        ...
        // Draw
        ...
    }
}
```