

Universidad Rey Juan Carlos

Gráficas y Visualización 3D

Práctica 1: *Draw a Point with a Mouse Click*

Katia Leal Algara

INTRODUCCIÓN:

Partiendo del código “Example 1_5.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 “01_click_points.html”. En este caso, se va a utilizar el buffer de datos para pasar la información a los shaders.

1. Definirse un `canvas` de 400x400.

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();  
        setupBuffers(); // Cambia un poco  
    } 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);};  
}
```

Como se puede observar, para registrar el controlador de los eventos de ratón, se utiliza la técnica de *función anónima*. De esta manera, mientras que desde el navegador se invoca a la función genérica asociada al controlador del ratón, la cual sólo recibe como parámetro el evento, gracias a la función anónima, en una segunda llamada, esta vez a la función *click*, podemos acceder a las variables locales de la función *main*.

4. La función *click()*

```
function click(ev, gl, canvas, vertexPositionAttribute) {  
    var x = ev.clientX; // x coordinate of a mouse pointer  
    var y = ev.clientY; // y coordinate of a mouse pointer  
    // ¿?  
    var rect = ev.target.getBoundingClientRect();  
  
    x = ((x - rect.left) - canvas.height/2)/(canvas.height/2);  
    y = (canvas.width/2 - (y - rect.top))/(canvas.width/2);  
    // Store the coordinates  
    mouseClickPoints.push(x);  
    mouseClickPoints.push(y);  
    // ¿?  
    // Clear <canvas>  
    gl.clear(gl.COLOR_BUFFER_BIT);  
    // Aquí viene el código para dibujar  
}
```

