Examen Gráficos y Visualización 3D

Se pide modificar **2 ejercicios de WebGL previamente entregados**, a elegir entre las siguientes 3 alternativas:

- Ejercicio 3 – Dibujar puntos de color haciendo click en canvas
- Ejercicio 4 – Transformaciones con WebGL
- Ejercicio 7 – Iluminación con WebGL

A continuación, se detallan los cambios a realizar en cada uno los ejercicios anteriores.

Recuerda que los ejercicios que usan texturas deben ser cargados a través de un servidor web. Puedes servir un directorio de tu máquina mediante un servidor web local utilizando la aplicación **Web Server for Chrome**, o ejecutando el siguiente comando (y luego visualizarlo en el navegador usando la URL http://localhost:8000/):

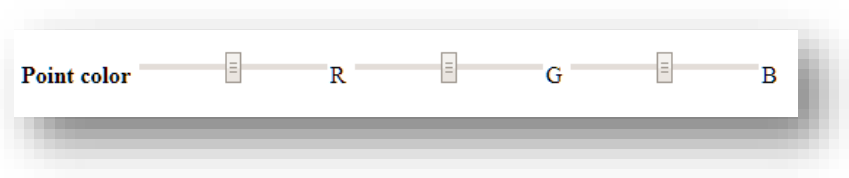
```
python -m SimpleHTTPServer
```

Se deberá **entregar un único fichero .zip** con los ejercicios modificados (y la textura).

Ejercicio 3 – Dibujar puntos de color haciendo click en canvas

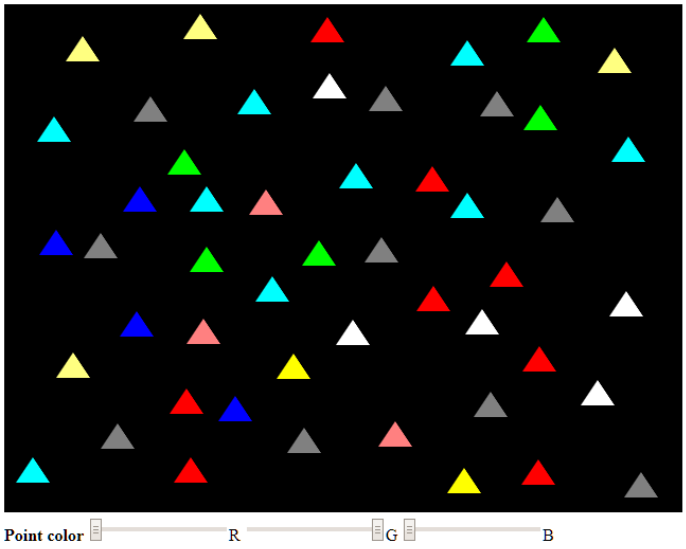
Partiendo del ejercicio 3, que consistía en dibujar puntos de color haciendo click en un canvas HTML, se pide hacer 2 modificaciones:

1. En el ejercicio original, el color de los puntos estaba determinado por la posición del click de ratón (rojo en el cuadrante superior derecha, verde en el cuadrante inferior derecha, azul en el cuadrante inferior izquierda, rosa en el cuadrante superior izquierda y blanco en los bordes). En la versión modificada, se pide que el color de los puntos se pueda elegir desde la interfaz de usuario mediante **campos de tipo rango**. Para ello se usarán tres controles de este tipo para elegir las componentes R (rojo), G (verde) y B (azul) del color de los puntos. Cada componente oscilará entre un valor mínimo de 0 y máximo de 1, con un paso (*step*) de 0.01. El valor por defecto de cada componente será de 0.5. La componente A (alfa) del color en cuestión será fija y de valor 1 (o sea, color opaco, sin transparencia).



2. En el ejercicio original, se dibujaba un punto de 10 o 20 píxeles por cada click de ratón. En la versión modificada, se debe dibujar **1 triángulo**. Suponiendo que las coordenadas del click son (x, y), los vértices del triángulo en coordenadas WebGL serán:
   (x, y+0.05, 0)
   (x+0.05, y-0.05, 0)
   (x-0.05, y-0.05, 0)

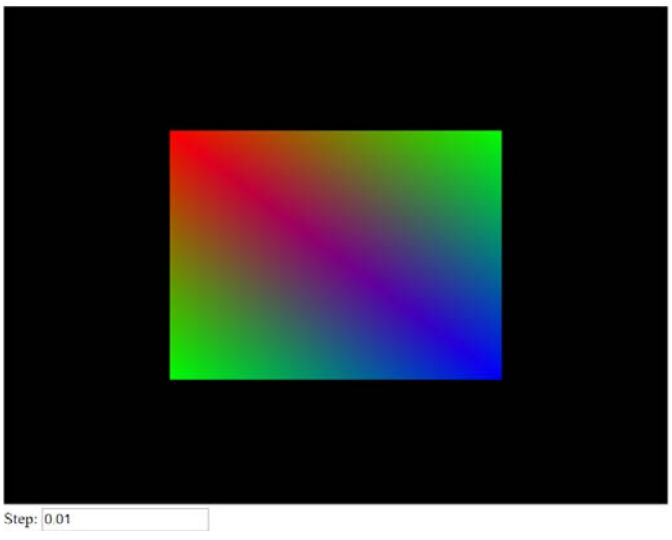
A continuación, se muestra un pantallazo a modo de ejemplo:



Ejercicio 4 – Transformaciones con WebGL

Partiendo del ejercicio 4, que consistía en realizar diferentes transformaciones (traslación, escalado y rotación) en un cuadrado en 2D mediante pulsaciones de teclas (flecha izquierda, derecha, etc.), se pide realizar los siguientes cambios:

1. El color original del rectángulo era amarillo. Se pide cambiar el color amarillo por un degradado de color en el que el vértice superior izquierdo será rojo, el vértice inferior izquierdo y el superior derecho, verde, y el vértice inferior derecho, azul. Tal y como se muestra en el siguiente pantallazo:



2. Se pide capturar la pulsación de la barra espaciadora (código de tecla **32**) para resetear todas las transformaciones realizadas. En otras palabras, después de mover el rectángulo, rotarlo, y hacerlo más grande o pequeño (escalado), cuando el usuario pulse espacio, el rectángulo volverá a su posición original y volverá a mostrarse sin rotar y sin escalar.

Ejercicio 7 – Iluminación con WebGL

Partiendo del ejercicio 7, que consistía en implementar una esfera con textura, y en el que además se podía seleccionar las coordenadas x, y, z de luz direccional, se pide realizar los siguientes cambios:

1. En el ejemplo original, la esfera giraba de forma indefinida. En la versión modificada, se capturará la pulsación de la barra espaciadora (código de tecla **32**) para parar el giro de la esfera. Cuando se vuelva a presionar la barra espaciadora, la esfera deberá seguir girando desde donde se quedó parada.
2. En el ejemplo original, la cámara estaba situada en las coordenadas x=0, y=0, z=3. En la versión modificada, se pide capturar las flechas Up (código **38**) y Down (código **40**) de forma que las flechas modifiquen la posición de la coordenada z de la cámara. Esto debe funcionar tanto en el caso de que la esfera esté girando, así como en el caso de que el giro de la esfera hubiese sido pausado previamente (mediante la barra espaciadora).