



## Ejercicios Tema 2 – parte 2

4.- Paso de atributos y variables uniform a los shader y su uso dentro de los shaders

Crea un nuevo proyecto e incluye en él los ficheros `tema2_parte2.cpp`, `tga.cpp`, `tga.h`, `vboteapot.cpp`, `vboteapot.h` y `teapotdata.h`

Crea un vertex shader con variables de entrada para recoger los atributos de la tetera (posición, normal y coord. de textura). Crea también en el vertex shader una variable uniform donde se recoja la matriz Modelo-Vista-Proyección, de manera que dentro del vertex shader se transforme la posición del vértice aplicando esta transformación. Por último, asigna como color al vértice las coordenadas de la normal en valor absoluto.

Crea el fragment shader y asigna el color al fragmento en función de los colores de los vértices al triángulo que pertenece.

Modifica en el programa las funciones `initTeapot`, `drawTeapot`, `init`, `display` para conseguir que se visualice la tetera.

Una vez hecho esto, realiza los siguientes cambios en la aplicación y en los shaders para crear los siguientes efectos:

- Que al pulsar la tecla 'P' las posiciones de los vértices se incrementen/decrementen para que la tetera se transforme de forma gradual en una esfera de radio 3.0 y luego vuelva de forma gradual a su estado original. El proceso se repetirá de forma periódica hasta que el usuario vuelva a pulsar la tecla 'P'.
- Que al pulsar la tecla 'R' los colores de los vértices de la tetera vayan rotando. Para conseguirlo, se le asignará como color al vértice la normal multiplicada por una matriz de rotación respecto al eje x, cuyo ángulo se incrementará en un grado en cada uno de los cuadros de la animación. El proceso se desactivará cuando el usuario vuelva a pulsar la tecla 'R'.
- Que al pulsar la tecla 'D' se descarten fragmentos (en el fragment shader) en función de las coordenadas de textura que tienen asignados (las coordenadas de textura s y t de los vértices de la tetera son valores comprendidos entre 0 y 1). Para decidir que fragmentos descartar, nos fijaremos en sus coordenadas de textura y realizaremos el siguiente proceso: multiplicamos esas coordenadas por 10 y nos quedamos con la parte decimal resultante. Si estos dos valores son simultáneamente mayores que un cierto valor umbral (comprendido entre 0 y 1) descartaremos el fragmento. El valor umbral variará entre 0 y 1 de forma sinusoidal en cada frame.



#### 5.- Texturas: multitexturas

Modifica el programa principal, el vertex shader y el fragment shader para conseguir los siguientes resultados:

- a) Que al pulsar la tecla 'M' se pegue sobre la tetera la textura 'stone.tga' (combinando los colores de los texels con los colores obtenidos en el ejercicio anterior).

Nota: la textura 'stone.tga' tiene 3 componentes (RGB).

- b) Que al volver a pulsar la tecla 'M' se dibuje con dos texturas (stone.tga y moss.tga), mezclando ambas texturas en función del valor del canal de transparencia de la última).

Nota: la textura 'moss.tga' tiene 4 componentes (RGBA).