Ejercicios Tema 2 – parte 1

1.- API de OpenGL para incluir shaders en la aplicación

Crea un proyecto en Microsoft Visual Studio e incluye en él el fichero tema2_ej1.cpp.

Vamos a modificar el anterior programa para adaptarlo a los requerimientos de las nuevas versiones de la OpenGL (que imponen el uso de shaders y evitan el uso de las funciones de la OpenGL marcadas como "obsoletas").

Para ello, en este ejercicio vamos cargar en la aplicación los shaders "`tema2_ej1.vert`" y "`tema2_ej1.frag`" (utilizando la función `loadSource` que aparece en el programa) y crearemos con ellos un objeto programa (todo ello dentro de la función `init`). Una vez hecho, modificaremos la función `display` para utilizar este objeto programa cada vez que se dibuja el cubo.

2.- Paso de atributos desde la aplicación

Modifica la primera línea del vertex sader y del fragment shader donde se indica la versión de la GLSL utilizada. En el código, el profile de la versión es "compatible", sustitúyelo por "core" y comprueba qué pasa.

A partir de la versión 1.40 de la GLSL desaparecen las variables globales especiales que recogían los atributos de los vértices (`gl_Vertex`, `gl_Color`, etc.). En su lugar, dentro del vertex shader se incluyen variables globales **`in`** que recogen estos atributos y los valores se pasan de forma explícita en la aplicación.

Modifica el programa y los shaders para eliminar las referencias a estas variables especiales.

3.- Paso de variables uniform desde la aplicación

A partir también de la versión 1.40 de la GLSL desaparecen las variables globales predefinidas que recogían el estado de la OpengGL, tales como la matriz de modelo-vista (`gl_ModelViewMatrix`) la matriz modelo-vista-proyección (`gl_ModelViewProjectionMatrix`), etc. Ello es debido, entre otras cosas, a que se declaran obsoletas en la OpenGL 3.x las pilas de matrices `GL_MODELVIEW` y `GL_PROJECTION`, así como todas las funciones del API de la OpengGL dedicadas a su manipulación (`glPushMatrix, glPopMatrix, glLoadIdentity, glRotate, glTranslate, gluPerspective`, etc.). En su lugar, dentro de los shaders se incluyen variables globales **`uniform`** que recogen esta información y los valores se pasan de forma explícita en la aplicación.

Modifica el programa para eliminar la referencia en el vertex shader a la variable predefinida `gl_ModelViewProjectionMatrix` y sustitúyela por una variable uniform, y pásale el valor de forma explícita en la aplicación.

Nota: Para esta parte necesitamos instalar en el ordenador la librería matemática “glm”. Esta librería matemática estaba incluida en el fichero de instalación “instalación.rar” que aparecía en Aula Virtual. Si quieres instalar la última versión disponible, ésta se encuentra en la url https://glm.g-truc.net/. Para instalarla, descomprime el fichero zip y copia la 2º carpeta glm en el directorio (donde XXXXX es la versión del Visual que tengas instalada):

C:\Archivos de programa (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\XXXXX\include