

**Lee detenidamente las instrucciones y el enunciado  
antes de empezar a trabajar!**

## Instrucciones

1. Puedes usar el código que has elaborado en las clases de laboratorio y que tengas en tu cuenta, pero **solamente el código que hayas generado tu**. No puedes usar código que otros estudiantes hayan compartido contigo ni que tu hayas compartido con otros. De lo contrario se considerará copia.
2. Partirás del código contenido en el fichero `examen.tgz` (adjunto a esta práctica). Tienes que descomprimirlo en un directorio tuyo. Se creará el subdirectorio `examen-1718Q2` donde encontrarás todos los ficheros con los que tienes que trabajar. Los ejercicios a resolver sólo requieren cambios en la clase `MyGLWidget`, en los `shaders` y en el fichero `MyForm.ui` usando el `designer`. **No tienes que modificar ningún otro fichero de los que se te proporcionan.**
3. **Si el código que entregas no compila o da error de ejecución, la evaluación será un cero**, sin excepciones.
4. Para hacer la entrega tienes que generar un archivo tar que incluya todo el código de tu examen y que se llame `<nombre-usuario>.tgz`, substituyendo `<nombre-usuario>` por tu nombre de usuario. Por ejemplo, el estudiante Pompeu Fabra (desde el directorio `examen-1718Q2`):

```
make distclean
tar zcvf pompeu.fabra.tgz *
```

Es importante el `'make distclean'` para borrar los archivos binarios generados; que el nombre de usuario sea el tuyo y que tenga el sufijo `.tgz`.

5. Una vez hecho esto, en tu directorio `examen-1718Q2` tendrás el archivo `<nombre-usuario>.tgz` que es lo que tienes que entregar. **Comprueba** que todo es correcto descomprimiéndolo **en un directorio vacío** i mirando que el código compila (haciendo `qmake-qt5; make`) y ejecuta correctamente.
6. Finalmente, entrega el fichero en <https://examens.fib.upc.edu>.

**Nota:** Recuerda que si abres el fichero `~/examen/assig/idi/man.3.3/index.html` desde el navegador tendrás acceso al manual de OpenGL 3.3, y que en `~/examen/assig/idi/glm/doc/api/index.html` tienes el manual de la librería `glm`. También tienes, como ya sabes, el `assistant-qt5` para dudas de Qt.

## Enunciado

El código que proporcionamos, tal y como está, pinta un suelo de  $20 \times 20$  unidades sobre el plano XZ centrado en el origen, con un Legoman de altura 1 y con el centro de la base de su caja contenedora en el origen de coordenadas (mira la imagen del fichero `EscIni.png`). La cámara está inicializada arbitrariamente y sólo se puede modificar interactivamente el ángulo  $\psi$ .

Observa que hay un método `createBuffers` para cada modelo, que se ha hecho para simplificar la lectura del código y su uso. Este método tiene inicializados todos los datos de material y normales para poder implementar el cálculo de la iluminación. También proporcionamos las rutinas `Lambert` y `Phong` que encontrarás en el Vertex Shader.

Este examen puntúa sobre 11 puntos. Se pueden conseguir 11 puntos si todo se hace perfectamente.

En la valoración de los ejercicios 4, 5 y 6 tendrá mucha importancia el diseño y la usabilidad de la interfaz.

1. (1.5 puntos) Modifica la escena dada para que, en lugar de un legoman haya 4 Patricios (modelo `Patricio.obj`): Pat1, Pat2, Pat3 y Pat4. El primer Patricio, Pat1, debe tener altura 2 y estará situado con el centro de su base en el punto (8, 0, 8). El segundo Patricio, Pat2, debe tener altura 4 y estará situado con el centro de su base en el punto (-8, 0, 8). El tercer Patricio, Pat3, debe tener altura 6 y estará situado con el centro de su base en el punto (8, 0, -8). Y el cuarto Patricio, Pat4, debe tener altura 8 y estará situado con el centro de su base en el punto (-8, 0, -8). Los cuatro Patricios tienen que estar escalados uniformemente y mirando hacia el eje Y de la escena (ver figura `EscSol11.png`). Recuerda que el Patricio inicialmente mira hacia Z+.

También debes modificar el material del suelo para que pase a ser de color verde brillante y con intensidad 0.6.

2. (2 puntos) Esta escena debe poder inspeccionarse con una cámara en tercera persona que permita inicialmente ver la escena centrada, entera, sin deformar y ocupando el máximo del viewport (siendo el viewport toda la ventana gráfica). La cámara debe tener una óptica perspectiva. En caso de redimensionamiento de la escena (resize) la escena no debe deformarse ni recortarse. Esta cámara también debe permitir la inspección mediante rotaciones en ángulos de Euler (ángulos  $\psi$  y  $\theta$ ), es decir, el usuario debe poder modificar estos ángulos usando el ratón como se ha hecho en el laboratorio. La cámara debe tener inicialmente ángulos  $\psi = M\_PI/4.0$  y  $\theta = 0$ .

Puedes ver una imagen de la solución a estos dos primeros ejercicios sin iluminación en `EscSol11.png`.

3. (1.5 puntos) Añade a la escena el cálculo de la iluminación **en el Vertex Shader** usando el modelo de iluminación de Phong y un foco de cámara de luz blanca situado siempre exactamente en la posición de la cámara.
4. (1.5 puntos) Define los elementos de interfaz que te permitan elegir uno de los 4 Patricios de la escena (Pat1 ... Pat 4). El Patricio que hayas seleccionado desaparece de la escena (no se pinta). Inicialmente, ningún Patricio está seleccionado y, una vez has seleccionado uno, se tiene que poder volver al estado en que ninguno está seleccionado (y por lo tanto se pintan los 4). En cualquier momento hay como máximo 1 Patricio seleccionado.

5. (2 puntos) Añade una segunda cámara que será una cámara en primera persona. Esta cámara se situará en la posición del Patricio seleccionado pero a la altura (coordenada Y) que corresponda al Patricio en cuestión. Por ejemplo: si el seleccionado es el Pat1, la cámara debería estar en la posición (8, 2, 8) porque Pat1 tiene su base en (8, 0, 8) y altura 2. Esta cámara tiene que estar orientada en la misma dirección en la que mira el Patricio y con vector up=(0,1,0). La óptica de esta cámara tiene que ser perspectiva con ángulo de abertura de  $M\_PI/2.0$  radianes (90 grados). Los valores de `Znear` y `Zfar` tienen que permitir que se vea completamente la escena que hay delante del patricio sobre el que se sitúa la cámara. La cámara no deformará la escena en caso de redimensionamiento del viewport.

Esta nueva cámara tiene que poder activarse/desactivarse con un elemento de la interfaz. Cuando se desactiva tiene que recuperarse la vista del ejercicio 2, y por lo tanto recuperarse los últimos valores que tuvo (ángulos  $\psi$  y  $\theta$ ). **No se podrá cambiar a esta cámara en primera persona si no hay ningún Patricio seleccionado.**

Si mientras esta cámara está activa se modifica la selección del Patricio, la posición y orientación de la cámara tendrán que modificarse inmediatamente para adaptarse a la nueva selección.

6. (1.5 puntos) Añade a tu interfaz un botón de "Reset" que devuelva a la situación inicial del programa. Es decir, la cámara está en tercera persona, con los parámetros iniciales descritos en el ejercicio 2, y ningún Patricio está seleccionado. Del mismo modo, todos los elementos de la interfaz implicados tienen que volver a sus valores iniciales.

7. (1 punto) **Ejercicio con puntuación extra.**

Modifica el mecanismo con el que muestras el Patricio seleccionado de manera que, en lugar de hacer que no se pinte, el seleccionado se pinte de color amarillo y sin iluminación.