

**Leed detenidamente las instrucciones y el enunciado  
antes de empezar a hacer nada!**

## Instrucciones

1. Podéis usar el código que habéis elaborado en las clases de laboratorio y que tengáis en vuestra cuenta, pero **sólo el código que hayáis generado vosotros**; no podéis usar código que otros estudiantes hayan compartido con vosotros (ni que hayáis compartido con otros estudiantes). De lo contrario se considerará copia.
2. Partiremos del código que tenéis en `examen.tgz` (adjunto a esta práctica). Tenéis que descomprimir este archivo en un directorio vuestro. Os creará un subdirectorio `examen` donde tendréis todos los ficheros con los que tenéis que trabajar. Los ejercicios que os pedimos sólo requieren cambios en la clase `MyGLWidget`, als `shaders` i al fitxer `MyForm.ui` usando el designer. **No tenéis que modificar ningún otro fichero de los que os damos.**
3. **Si el código que entreguéis no compila o da error de ejecución, la evaluación será un 0**, sin excepciones.
4. Para hacer la entrega tenéis que generar un archivo `.tar` que incluya todo el código de vuestro examen que se llame `<nombre-usuario>.tgz`, donde sustituiréis `<nombre-usuario>` por vuestro nombre de usuario. Por ejemplo, el estudiante Pompeu Fabra (desde una terminal en la que se ha colocado dentro del directorio `examen`):

```
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 correcto (el vuestro); y que tenga el sufijo `GL.tgz`

5. Una vez hecho esto, en vuestro directorio `examen` tendréis el archivo `<nombre-usuario>.tgz` que es lo que tenéis que entregar. **Haced la comprobación**, descomprimiendo este archivo **en un directorio completamente vacío**, que el código que entreguéis compila (haciendo `qmake-qt5; make`) y ejecuta correctamente.
6. Finalmente, entregad el fichero en <https://examens.fib.upc.edu>.

**Nota:** Recordad que si abris el fichero `/assig/idi/man_3.3/index.html` desde el Firefox o el Konqueror tendréis acceso a las páginas del manual de OpenGL 3.3, y con el fichero `/assig/idi/glm/doc/api/index.html` tendréis acceso a las páginas del manual de la librería glm. También tenéis, como ya sabéis, el `assistant` para dudas de Qt.

## Enunciado

El código que os pasamos tal y como está pinta un suelo de 6x6 unidades sobre el plano XZ y centrado en el punto  $(-2,0,0)$ , con un sofá en un lado de altura 1 que se encuentra en una dirección paralela al eje de las Z y un legoman de altura 1 con el centro de la base de su caja contenedora en el origen (ver una imagen en el archivo `EscIni-30.png`). La escena se pinta con una cámara arbitraria a la que sólo le podemos modificar interactivamente el ángulo  $\psi$ .

El código también tiene inicializados todos los datos de material y normales necesarios para poder implementar el cálculo de la iluminación. También os pasamos las rutinas **Lambert** y **Phong** que se encuentran en el Vertex Shader.

Fijaos que hay un método **createBuffers** para cada modelo, que se ha hecho para simplificar la lectura del código y su utilización.

1. (3 puntos) Modifica esta escena para que el legoman sea de altura 2 (escalado uniformemente), esté situado con el centro de su base en el punto (1, 0, 0) y mirando en dirección X- (hacia el sofá).

Esta escena se tiene que poder inspeccionar con una cámara en tercera persona que permita ver la escena entera, centrada, sin recortar ni deformar y ocupando el máximo del viewport. La cámara debe tener una óptica en perspectiva. En caso que se redimensione la ventana (resize) la escena no se tiene que recortar ni deformar. Esta cámara inicial también tiene que permitir la inspección mediante rotaciones de los ángulos de Euler (ángulos  $\psi$  y  $\theta$ ), o sea, el usuario tiene que poder modificar estos ángulos usando el ratón como hemos visto en el laboratorio. La cámara inicialmente tiene que estar mirando hacia el legoman y viendo por lo tanto el sofá desde atrás. La altura de esta cámara inicial respecto al centro de la escena tiene que ser 0.

Una imagen de la solución de este ejercicio la podéis ver en el archivo **EscSol1-30.png**.

2. (1,5 puntos) Añade a la escena el cálculo de la iluminación **en el Fragment Shader** usando el modelo de iluminación de Phong y con un foco de cámara blanco situado en la posición de la cámara.
3. (2,5 puntos) Haz que cada vez que el usuario pulse la tecla de la flecha hacia arriba **Key\_Up** el legoman se mueva 0,5 unidades en dirección X- (hacia el sofá). Cuando el legoman llegue al sofá (posición del centro de la base con componente  $x == -2.5$ ), es necesario pintar en el su lugar un legoman-assegut (modelo legoman-assegut.obj), que será igualmente de altura 2, tendrá el centro de su base situado en el punto (-2.5, 0.5, 0) y estará mirando en dirección X+ (la inversa del que teníamos).

Fijaos que en vez de cargar cada vez un nuevo modelo (sustituyendo el anterior que estaba cargado), podéis también tener los dos modelos cargados a la vez (una sola vez cada uno) y decidir en cada momento cual pintáis. Esta segunda opción sería de hecho la más eficiente.

Pulsando la tecla I se tiene que poder recuperar el modelo inicial de legoman en su posición inicial.

4. (2 puntos) Añade a la interfaz 3 checkboxes etiquetados con las etiquetas "Red", "Green" y "Blue" que permitirán al usuario modificar el color (componente a componente y sólo con valores 0 o 1) del foco de luz. Inicialmente estos checkboxes tienen que estar los tres activados (porque el foco que tenemos inicialmente es blanco).
5. (1 punto) Añade un botón de **Reset** en la interfaz de manera que cuando se pulse el botón, la aplicación vuelva a los valores iniciales de la cámara del ejercicio 1 y del color del foco de luz.