

**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 hayais generado vosotros**; no podéis compartir código con los otros alumnos. 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. **No tenéis que modificar los ficheros `examen.pro` ni el `main.cpp`**. Los ejercicios que os pedimos sólo requieren cambios en la clase `MyGLWidget` y en los shaders.
3. **Si el código que entreguéis no compila, 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-usuari>_GL.tgz`, donde sustituiréis `<nombre-usuari>` 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_GL.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 la raya baja `'_'` separando el nombre de usuario del sufijo `GL.tgz`

5. Una vez hecho esto, en vuestro directorio `examen` tendréis el archivo `<nombre-usuario>_GL.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; make`) y ejecuta correctamente.
6. Finalmente, entregad el fichero en <https://examens.fib.upc.edu>

**Nota:** Recordad que si abríis 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 `/usr/include/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 20x20 centrado en el origen y un Patricio sin escalar situado con el centro de su base en el origen de coordenades (ver una imagen en el fichero `EscIni-20.png`). Tiene inicializados todos los datos de materiales y normales necesarios para poder hacer el cálculo de la iluminación. También os pasamos los métodos `Lambert` y `Phong` que se encuentran en el Vertex Shader.

1. (3 puntos) Modifica esta escena para que en lugar de un Patricio sin escalar y con la base centrada en el origen, haya un Patricio de altura 3 con la base centrada en el punto (10, 0, -10) (esquina del suelo) mirando hacia el eje vertical que pasa por el centro del suelo, y un legoman (modelo `legoman.obj`), también de altura 3 con la base centrada en el punto (0, 0, 0) mirando hacia el Patricio.

Esta escena se debe ver centrada y sin recortar, y aprovechando el máximo del viewport (vista), con una cámara perspectiva. En caso de redimensionamiento de la ventana (resize) no es necesario considerar el caso en el que la escena se recorte, pero en ningún caso la escena se puede deformar.

Una imagen posible de la solución a este ejercicio la podéis ver en el archivo `EscSol11-20.png`.

2. (2.5 puntos) Añade a la escena el cálculo de iluminación en el Fragment Shader usando el modelo de Phong y con un foco de luz blanca en la posición de la cámara.
3. (3 puntos) Haz que al pulsar la tecla 'C' todos los objetos de la escena se pinten con un material azul brillante. Al pulsar de nuevo la tecla 'C' se vuelve de nuevo a los materiales originales.
4. (1.5 puntos) Haz que el legoman gire  $M_{PI}/6.0$  radianes respecto a el eje de las Y cada vez que el usuario apriete la tecla 'R'.
5. (1.5 puntos) Haz que al pulsar la tecla 'L' el foco de luz cambie a la posición (10,10,10) encima del Patricio. Al apretar la tecla 'L' de nuevo, volverá a la posición de cámara original.