

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 usar código que otros estudiantes hayan compartido con vosotros (ni que hayais 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. **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 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-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 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 `/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 coordenadas, y con una cámara inicializada de forma arbitraria (ver una imagen en el fichero `EscIni-10.png`). Tiene inicializados todos los datos de materiales y normales necesarios para poder implementar 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 suelo, haya 4 Patricios de altura 3 situados en círculo mirando todos hacia el centro, es decir, un primer Patricio tendrá el centro de su base en el punto $(-1.5, 0, 0)$ y estará mirando en dirección $X+$, el segundo Patricio tendrá el centro de su base en el punto $(0, 0, 1.5)$ y estará mirando en dirección $Z-$, el tercero tendrá el centro de su base en el punto $(1.5, 0, 0)$ y estará mirando en dirección $X-$ y el cuarto y último tendrá el centro de su base en el punto $(0, 0, -1.5)$ y estará mirando en dirección $Z+$.

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) la escena no debe recortarse ni deformarse.

Una imagen posible de la solución a este ejercicio la podéis ver en el archivo `EscSo11-10.png`.

2. (2 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 que siempre esté en la posición de la cámara.
3. (2 puntos) Haz que cada vez que se aprete la tecla 'R', el círculo de Patricios (los 4 Patricios) giren 30 grados ($M_PI/6.0$ radianes) alrededor del eje Y de la escena, es decir, la primera vez que se apreta la tecla 'R' los 4 Patricios deben girar 30 grados respecto a la posición inicial, la segunda vez deben girar 30 grados más respecto de la posición anterior y así sucesivamente. Este giro del círculo de Patricios debe funcionar independientemente de la cámara que esté activa.
4. (2 puntos) Haz que cuando se aprete la tecla 'C' se cambie de cámara de forma que pasemos a tener una cámara justo en el centro del círculo de Patricios, a altura 1.5 y mirando hacia el eje $Z+$. Esta segunda cámara debe ser una cámara perspectiva con ángulo de apertura de $M_PI/2.0$ radianes (90 grados), posición y orientación fija y Znear y Zfar adecuados para ver todo lo que se pueda ver de la escena desde su posición y en aquella dirección. La cámara no debería deformar la escena en caso de redimensionamiento del viewport.

Apretando de nuevo la tecla 'C' se debe poder recuperar la cámara inicial o anterior.

Para la posición y orientación de esta segunda cámara puedes usar tanto la rutina `lookAt` como transformaciones geométricas, lo que prefieras.

5. (1 punto) Haz que cuando se aprete la tecla 'M' el material del suelo pase a ser azul brillante. Si se vuelve a apretar 'M' vuelve a ser el material inicial y así sucesivamente.