

Llegiu detingudament les instruccions i l'enunciat abans de començar a fer res!

Instruccions

1. Podeu usar el codi que heu elaborat en les classes de laboratori i que tingueu al vostre compte, però **sols el codi que hagueu generat vosaltres**; no podeu compartir codi amb els altres alumnes. Altrament es considerarà còpia.
2. Partirem del codi que teniu a `examen.tgz` (adjunt a aquesta pràctica). Heu de desplegar aquest arxiu en un directori vostre. Us crearà un subdirectori `examen` on tindreu tots els fitxers amb els que heu de treballar. **No heu de modificar els fitxers `examen.pro` ni el `main.cpp`**. Els exercicis que es demanen només requereixen canvis a la classe `MyGLWidget` i als shaders.
3. **Si el codi que entregueu no compila, l'avaluació serà un 0**, sense excepció.
4. Per a fer l'entrega heu de generar un arxiu tar que inclogui tot el codi del vostre examen que es digui `<nom-usuari>_GL.tgz`, on substituireu `<nom-usuari>` pel vostre nom d'usuari. Per exemple, l'estudiant Pompeu Fabra (des d'una terminal en la que s'ha col·locat dins del directori `examen`):

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

És important el `'make distclean'` per a esborrar els arxius binaris generats; que el nom d'usuari sigui el correcte (el vostre); i que hi hagi la ratlla baixa `'_'` separant el nom d'usuari del suffix `GL.tgz`

5. Un cop fet això, al vostre directori `examen` tindreu l'arxiu `<nom-usuari>_GL.tgz` que és el que heu d'entregar. **Feu la comprovació**, desplegant aquest arxiu **en un directori completament buit**, que el codi que entregueu compila (fent `qmake; make`) i executa correctament.
6. Finalment, lliureu el fitxer a <https://examens.fib.upc.edu>

Nota: Recordeu que si obriu el fitxer `/assig/idi/man_3.3/index.html` des del Firefox o el Konqueror tindreu accés a les pàgines del manual d'OpenGL 3.3, i amb el fitxer `/usr/include/glm/doc/api/index.html` tindreu accés a les pàgines del manual de la llibreria glm. També teniu, com bé sabeu, l'`assistant` per a dubtes de Qt.

Enunciat

El codi que us passem tal i com està pinta un terra de 20x20 centrat a l'origen i un Patricio sense escalar situat amb el centre de la seva base a l'origen de coordenades (veure una imatge a l'arxiu `EscIni-20.png`). Té inicialitzades totes les dades de material i normals necessàries per a poder fer el càlcul de la il·luminació. També us passem les rutines **Lambert** i **Phong** que es troben al Vertex Shader.

1. (*3 punts*) Modifica aquesta escena per a que en lloc d'un Patricio sense escalar i amb la base centrada a l'origen, hi hagi un Patricio d'alçada 3 amb la base centrada al punt (10, 0, -10) (cantonada del terra) mirant cap a l'eix vertical que passa pel centre del terra, i un legoman (model `legoman.obj`), també d'alçada 3 amb la base centrada al punt (0, 0, 0) mirant cap al Patricio.

Aquesta escena s'ha de veure centrada i sense retallar, i aprofitant el màxim del viewport (vista), amb una càmera perspectiva. En cas de redimensionament de la finestra (resize) no cal considerar el cas en que l'escena es retalli, però en cap cas l'escena es pot deformar.

Una imatge possible de la solució a aquest exercici la podeu veure a l'arxiu `EscSol1-20.png`.

2. (2.5 punts) Afegeix a l'escena el càlcul d'il·luminació al Fragment Shader usant el model de Phong i amb un focus de llum blanca a la posició de la càmera.
3. (1.5 punts) Fes que en prémer la tecla 'C' tots els objectes de l'escena es pintin amb un material blau brillant. En prémer la tecla 'C' de nou torna als materials originals.
4. (1.5 punts) Fes que el legoman roti $M_{PI}/6.0$ radians respecte l'eix de les Y cada cop que l'usuari prem la tecla 'R'.
5. (1.5 punts) Fes que en prémer la tecla 'L' el focus de llum canviï a la posició (10,10,10) al damunt del Patricio. En prémer la tecla 'L' de nou, tornarà a la posició de càmera original.