

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 fer servir codi que altres estudiants hagin compartit amb vosaltres (ni que hagueu compartit amb d'altres estudiants). 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. Els exercicis que es demanen només requereixen canvis a la classe `MyGLWidget`, als `shaders` i al fitxer `MyForm.ui` usant el designer. **No heu de modificar cap altre fitxer dels que us donem.**
3. **Si el codi que entregueu no compila o dóna error d'execució, 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>.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.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 el sufix `.tgz`

5. Un cop fet això, al vostre directori `examen` tindreu l'arxiu `<nom-usuari>.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-qt5; 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 `/assig/idi/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 6x6 unitats sobre el pla XZ i centrat al punt $(-2, 0, 0)$, amb un sofà en un costat d'alçada 1 que es troba en una direcció paral·lela a l'eix de les Z i un legoman d'alçada 1 amb el centre de la base de la seva capsa contenidora a l'origen (veure una imatge a l'arxiu `EscIni-30.png`). L'escena es pinta amb una càmera arbitrària a la qual només li podem modificar interactivament l'angle ψ .

El codi també té inicialitzades totes les dades de material i normals necessàries per a poder implementar el càlcul de la il·luminació. També us passem les rutines `Lambert` i `Phong` que es troben al Vertex Shader.

Fixeu-vos que hi ha un mètode `createBuffers` per a cada model, que s'ha fet per a simplificar la lectura del codi i la seva utilització.

1. (3 punts) Modifica aquesta escena per a que el legoman sigui d'alçada 2 (escalat uniformement), estigui situat amb el centre de la seva base al punt (1, 0, 0) i mirant en direcció X- (cap al sofà).

Aquesta escena s'ha de poder inspeccionar amb una càmera en tercera persona que permeti veure l'escena sencera, centrada, sense retallar ni deformar i ocupant el màxim del viewport. La càmera ha de tenir una òptica perspectiva. En cas de redimensionament de la finestra (resize) l'escena no s'ha de retallar ni deformar. Aquesta càmera inicial també ha de permetre la inspecció mitjançant rotacions dels angles d'Euler (angles ψ i θ), és a dir l'usuari ha de poder modificar aquests angles utilitzant el ratolí com hem vist al laboratori. La càmera inicialment ha d'estar mirant cap al legoman i veient per tant el sofà des del darrere. L'alçada d'aquesta càmera inicial respecte al centre de l'escena ha de ser 0.

Una imatge de la solució a aquest exercici la podeu veure a l'arxiu **EscSol1-30.png**.

2. (1.5 punts) Afegeix a l'escena el càlcul d'il·luminació **al Fragment Shader** usant el model d'il·luminació de Phong i amb un focus de càmera blanc situat a la posició de la càmera.
3. (2.5 punts) Fes que cada cop que l'usuari premi la tecla de la fletxa cap amunt **Key_Up** el legoman es mogui 0.5 unitats en direcció X- (cap al sofà). Quan el legoman arriba al sofà (posició del centre de la base amb component $x == -2.5$), cal pintar en el seu lloc un legoman-assegut (model legoman-assegut.obj), que serà igualment d'alçada 2, tindrà el centre de la seva base situat al punt (-2.5, 0.5, 0) i estarà mirant en direcció X+ (a la inversa del que teníem).

Fixeu-vos que en lloc de carregar cada vegada un nou model (substituint l'anterior que estava carregat), podeu també tenir els dos models carregats a la vegada (un sol cop cadascun) i decidir en cada moment quin pinteu. Aquesta segona seria de fet l'opció més eficient.

Prement la tecla I s'ha de poder recuperar el model inicial del legoman a la seva posició inicial.

4. (2 punts) Afegeix a la interfície 3 checkboxes etiquetats amb les etiquetes "Red", "Green" i "Blue" que permetran a l'usuari modificar el color (component a component i només amb valors 0 o 1) del focus de llum. Inicialment aquests checkboxes han d'estar tots tres activats (perquè el focus que tenim inicialment és blanc).
5. (1 punt) Afegeix un botó de **Reset** a la interfície de manera que quan es premi el botó, l'aplicació retorni als valors inicials de la càmera de l'exercici 1 i del color del focus de llum.