Examen de laboratori d'IDI 2016-2017, Q2. Temps: 1h 45'

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

Instruccions

1. Pots usar el codi que has elaborat en les classes de laboratori i que tinguis al teu compte, però **sols el codi que hagis generat tu**; no pots fer servir codi que altres estudiants hagin compartit amb tu (ni que tu hagis compartit amb d'altres estudiants). Altrament es considerarà còpia.

2. Partiràs del codi que tens a `examen.tgz` (adjunt a aquesta pràctica). Has de desplegar aquest arxiu en un directori teu. Es crearà un subdirectori `examen-1617Q2` on tindràs tots els fitxers amb els que has 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 has de modificar cap altre fitxer dels que et donem**.

3. **Si el codi que entregues no compila o dóna error d'execució, l'avaluació serà un 0**, sense excepció.

4. Per a fer l'entrega has de generar un arxiu tar que inclogui tot el codi del teu examen i que es digui `<nom-usuari>.tgz`, on substituiràs `<nom-usuari>` pel teu nom d'usuari. Per exemple, l'estudiant Pompeu Fabra (des d'una terminal en la que s'ha col·locat dins del directori `examen-1617Q2`):

   ```
   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 teu); i que hi hagi el sufix `.tgz`

5. Un cop fet això, al teu directori `examen-1617Q2` tindràs l'arxiu `<nom-usuari>.tgz` que és el que has d'entregar. **Fes la comprovació**, desplegant aquest arxiu **en un directori completament buit**, que el codi que entregues compila (fent `qmake-qt5; make`) i executa correctament.

6. Finalment, lliura el fitxer a `https://examens.fib.upc.edu`

Nota: Recorda que si obres el fitxer `~/examen/assig/idi/man_3.3/index.html` des del navegador tindràs accés a les pàgines del manual d'OpenGL 3.3, i amb `~/examen/assig/idi/glm/doc/api/index.html` tindràs accés a les pàgines del manual de la llibreria glm. També tens, com bé saps, l'`assistant` per a dubtes de Qt.

Enunciat

El codi que proporcionem, tal i com està pinta un terra de 20×20 unitats sobre el pla XZ i centrat a l'origen, amb un Legoman d'alçada 1 i el centre de la base de la seva capsa contenidora a l'origen de coordenades (mira la imatge de l'arxiu `EscIni.png`). L'escena es pinta amb una càmera arbitrària a la qual només es pot 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 proporcionem les rutines `Lambert` i `Phong` que es troben al Vertex Shader.

Fixa't que hi ha un mètode *createBuffers* per a cada model, que s'ha fet per simplificar la lectura del codi i la seva utilització.

1. *(2 punts)* Modifica l'escena donada per a que, en lloc d'un legoman tal i com està, hi hagi 4 Patricios (model `Patricio.obj`) d'alçada 5 (escalats uniformement) que estiguin situats en rotllana mirant tots cap al centre. En concret: un primer Patricio tindrà el centre de la seva base al punt (-2.5, 0, 0) i estarà mirant en direcció X+, el segon Patricio tindrà el centre de la seva base al punt (0, 0, 2.5) i estarà mirant en direcció Z-, el tercer tindrà el centre de la seva base al punt (2.5, 0, 0) i estarà mirant en direcció X- i el quart i últim tindrà el centre de la seva base al punt (0, 0, -2.5) i estarà mirant en direcció Z+. Recorda que el Patricio inicialment mira cap a les Z+.

   També has de modificar el material del terra per a que passi a ser un material brillant (no s'ha de canviar el color).

2. *(1.5 punts)* Aquesta escena s'ha de poder inspeccionar amb una càmera en tercera persona que permeti inicialment veure l'escena sencera, centrada, sense 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 deformar. Aquesta càmera 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 en direcció paral·lela a l'eix Z- de l'aplicació.

   Una imatge de la solució a aquests dos primers exercicis la pots veure a l'arxiu `EscSol1.png`.

3. *(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 de llum blanca situat sempre exactament a la posició de la càmera.

4. *(1.5 punts)* Fes que la rotllana dels 4 Patricios giri al voltant de l'eix Y de l'escena. L'angle de gir vindrà donat per un dial que cal afegir a la interfície. El dial ha de tenir valors en el rang [0..359] graus, de manera que en fer una volta al dial, els 4 Patricios (la rotllana de Patricios) fan una volta sencera al voltant de l'eix Y de l'escena. El valor inicial d'aquest angle de gir de la rotllana ha de ser 0.

   El gir de la rotllana de Patricios ha de funcionar independentment de la càmera que estigui activa.

5. *(2 punts)* Afegeix una segona càmera fixada a la posició (0, 2.5, 0) de l'escena i mirant cap a l'eix Z+. L'òptica d'aquesta càmera ha de ser perspectiva amb angle d'obertura de M_PI/2.0 radians (90 graus) amb posició i orientació fixes. Els valors de `Znear` i `Zfar` permetran veure tot el que es pugui veure de l'escena des de la posició de la càmera i en la direcció indicada. La càmera no deformarà l'escena en cas de redimensionament del viewport.

   La nova càmera i la programada a l'Exercici 2 podran intercanviar-se prement la tecla `C`.

   Per a la posició i orientació d'aquesta segona càmera pots usar, si vols, la crida `lookAt` de la glm.

6. *(1.5 punts)* Afegeix a la interfície un slider que permeti modificar el canal verd (G) del color (RGB) del focus de llum. Inicialment aquest slider ha d'estar al seu valor màxim perquè el focus és blanc (component G=1), però l'usuari ha de poder modificar el seu valor, amb la qual cosa el color del focus passarà a tenir menys component verda fins que arribi al valor mínim de l'slider (en aquest valor mínim la component G haurà de ser 0).