Examen d'OpenGL d'IDI 2015-2016, Q2. Grup 30. Temps: 1h 30'

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 gml. 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 4x4 unitats sobre el pla XZ i centrat a l'origen i un Patricio sense escalar amb el centre de la base de la seva capsa contenidora a l'origen de coordenades. 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.

1. *(4 punts)* Modifica aquesta escena per a que, en lloc del que hi ha, hi hagi un terra de mida 10x10 centrat a l'origen i amb 2 Patricios de la manera següent: El primer Patricio ha de tenir alçada 1 (escalat uniformement) i ha d'estar situat amb el centre de la seva base al punt (-4, 0, 4) i estarà mirant en direcció cap a l'eix Y de l'aplicació, el segon Patricio serà d'alçada 1.5 i ha de tenir el centre de la seva base al punt (-4, 0, -4) i també estarà mirant en direcció cap a l'eix Y de l'aplicació.

   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) l'escena no s'ha de retallar ni deformar.

   Aquesta càmera inicial també ha de permetre inspecció mitjançant rotacions dels angles d'Euler (angles ψ i θ), és a dir l'usuari ha de poder modificar aquests angles amb el ratolí.

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

2. *(1 punt)* Afegeix a l'escena el càlcul d'il·luminació **al Vertex Shader** usant el model d'il·luminació de Phong i amb un focus de llum blanca situat a la posició (5, 3, 5) de l'escena.

3. *(1.5 punts)* Fes que usant les tecles 'D' i 'A' el segon Patricio es mogui sobre la diagonal del terra augmentant/disminuint la seva posició sempre en (0.5, 0, 0.5) i rotant sobre l'eix Y del propi model M_PI/8 de forma incremental/decremental en cada pas. És a dir, quan l'usuari prem la tecla 'D', la posició del Patricio s'augmenta amb el vector (0.5, 0, 0.5) i l'angle s'incrementa en M_PI/8. Mentre que si la tecla és al 'A', la posició es disminueix amb el vector (-0.5, 0, -0.5) i l'angle de rotació es decrementa en M_PI/8.

4. *(2 punts)* Fes que en prémer la tecla 'C' es canviï de càmera de manera que ara passem a tenir una càmera just a sobre del primer Patricio, a alçada 1.5 i mirant **sempre** cap al centre del segon Patricio (també quan aquest es mou). Aquesta segona càmera ha de ser una càmera perspectiva amb angle d'obertura de M_PI/2.0 radians (90 graus), posició fixa (sense interacció) i Znear i Zfar adients per a veure tot el que es pugui veure de l'escena des de la seva posició. La càmera no hauria de deformar l'escena en cas de redimensionament del viewport.

   Prement de nou la tecla 'C' s'ha de poder recuperar la càmera inicial o anterior.

   Per a la posició i orientació d'aquesta segona càmera pots usar tant la crida lookAt com transformacions geomètriques, el que prefereixis.

5. *(1.5 punts)* Fes que en prémer la tecla 'L' el focus passi a ser de càmera, a la posició de la càmera i de color vermell. Aquest canvi en el focus s'ha de decidir en el Vertex Shader en funció d'un uniform que li indicarà quan s'ha de pintar amb aquest nou focus (que el podeu afegir *hard-coded* en el shader) i quan s'ha de pintar amb el focus original. Si es torna a prémer 'L' s'ha de tornar a pintar tot amb el color del focus original i així successivament.