

## 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 hakis generat tu**; no pots fer servir codi que altres estudiants hagin compartit amb tu (ni que tu hakis 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-1819Q1` on tindràs tots els fitxers amb els quals 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.**
3. **Si el teu codi 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 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-1819Q1`):

```
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-1819Q1` 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:** 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 l'`assistant-qt5` per a dubtes de Qt.

### Enunciat

El codi que proporcionem crea i visualitza una escena formada per un camp de futbol de 20x12 unitats ubicat sobre el pla XZ i centrat a l'origen, una porteria d'alçada 6 a un costat del camp, una pilota amb el centre de la base de la seva capsula contenidora al punt (6, 0, 0) i un Patricio. La càmera està inicialitzada arbitràriament i només es pot modificar interactivament l'angle  $\psi$ . La imatge de l'arxiu `EscIni.png` mostra la visualització inicial de l'escena.

Hi ha un mètode `createBuffers` per a cada model. Aquest mètode té inicialitzades totes les dades de material i normals necessàries per poder implementar el càlcul de la il·luminació. També proporcionem les rutines Lambert i Phong que es troben al Vertex Shader. També **es dóna ja implementat el moviment bàsic de la pilota**, de manera que si es prem la tecla `Key_Up` (xutem) la pilota es posa en moviment fins que arriba al fons de la porteria. Mitjançant la tecla `Key_I` la pilota torna al seu punt d'inici i es pot tornar a xutar. **Observació: Analitzeu el codi donat abans d'implementar funcionalitats.**

En la valoració dels exercicis 4, 5 i 6 tindrà molta importància el disseny i la usabilitat de la interfície.

1. (1.5 punts) Modifica l'escena per a que, en lloc d'un Patricio hi hagi 2 legomans (model `legoman.obj`): Lego1, Lego2. Tots dos legomans han de tenir alçada 4. El primer legoman, Lego1, és el jugador que xuta i estarà situat amb el centre de la base de la seva capsa contenidora al punt (8, 0, 0) i mirant en direcció X-. El segon legoman, Lego2, serà qui farà de porter i estarà situat inicialment amb el centre de la base de la seva capsa al punt (-7, 0, 0) i mirant en direcció X+. Observació: el model `legoman.obj` mira en direcció Z+.
2. (2 punts) L'escena s'ha de poder inspeccionar amb una càmera en tercera persona que permeti veure l'escena centrada, sencera, sense deformar i ocupant el màxim del viewport (essent el viewport tota la finestra gràfica). La càmera ha de tenir una òptica perspectiva. En cas de redimensionament de la finestra (resize) l'escena no s'ha de deformar ni retallar. Aquesta càmera també ha de permetre la inspecció mitjançant rotacions dels angles d'Euler (angles  $\psi$  i  $\theta$ ), és a dir l'usuari ha de poder modificar aquests angles utilitzant el ratolí com s'ha fet al laboratori. La càmera inicial ha de tenir angles  $\psi = M\_PI/4.0$  i  $\theta = 0$ .

Una imatge de la solució a aquests 2 primers exercicis la tens a `EscSol1.png`.

3. (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 sempre exactament a la posició de la càmera.
4. (1.5 punts) Afegeix una segona càmera que serà una càmera en primera persona. Aquesta càmera se situarà a la posició (8, 5, 0) i ha de mirar en la mateixa direcció en què mira el Lego1 i amb el vector up (0,1,0). L'òptica d'aquesta càmera ha de ser perspectiva amb angle d'obertura fix de  $M\_PI/2.0$  radians (90 graus). Els valors de `Znear` i `Zfar` han de permetre veure la part de l'escena que hi ha al davant del legoman. La càmera no es modificarà amb interacció del ratolí i no deformarà l'escena en cas de redimensionament del viewport.

Aquesta nova càmera s'ha de poder activar/desactivar mitjançant un element d'interfície. Quan es desactiva cal tornar a la vista anterior de la càmera en tercera persona.

5. (1.5 punts) Per a poder modificar la trajectòria de la pilota i fer més visible quan es fa un gol cal que feu el següent:
  - (a) Defineix un element a la interfície que permeti indicar/modificar l'angle en què la pilota es xuta. Aquest angle inicialment és 0 i s'ha de poder variar entre -15 i 15 graus. Fixa't que aquest angle el rep el mètode `mouPilota(int alfa)` en graus, que ja el teniu implementat.
  - (b) Quan la pilota entra a porteria cal que el focus de llum passi a ser de color vermell. Implementa el mètode `tractamentGol()` per a què faci aquest canvi de color del focus.
  - (c) Quan l'usuari torna la pilota a l'inici (amb la tecla `Key_I`) cal que el focus torni a ser blanc.
6. (1 punt) Afegeix la possibilitat que el legoman porter, Lego2, es mogui en la direcció de l'eix Z, entre els punts `pmin=(-7,0,-3)` i `pmax=(-7,0,3)`. Fes que el moviment vingui marcat per les fletxes laterals (`Key_Left` i `Key_Right`) de manera que l'esquerra el faci moure en direcció Z+ i la dreta el faci moure en direcció Z- (corresponen a esquerra i dreta vist des del punt de vista del Lego1).

Completa el mètode `aturaPorter()` fent que comprovi si el porter ha aturat la pilota. Per a comprovar aquesta aturada, cal que es compleixin les dues següents condicions:

$$posPilota.z \in [posPorter.z - 1, posPorter.z + 1]$$

$$posPilota.x \in [-6.3, -5.8]$$

7. (1.5 punts) Afegeix a la teva interfície un botó de "Reset" que ha de tornar a la situació inicial del programa. És a dir, ha d'estar en càmera en tercera persona, amb els paràmetres descrits a l'exercici 2, la pilota i el porter han d'estar també a la posició inicial i amb l'angle inicial i el color del focus a blanc. De la mateixa manera tots els elements d'interfície implicats han de tornar també als valors inicials.