

## 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-1819Q2` 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 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-1819Q2`):

```
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-1819Q2` 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 dels navegadors firefox o konqueror 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 joc de 20x14 unitats ubicat sobre el pla XZ i centrat a l'origen, una pilota amb el centre de la base de la seva capsula contenidora al punt (7, 0, 0), un cub de costat 1 amb el centre de la seva base a l'origen de coordenades 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 `creaBuffers` 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` la pilota es posa en moviment fins que arriba a un límit fora del camp de joc. Mitjançant la tecla `Key_I` la pilota torna al seu punt d'inici i es pot tornar a posar en moviment. **Observació: Analitzeu el codi donat abans d'implementar funcionalitats.**

En la valoració dels exercicis 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 1 legoman (model `legoman.obj`) d'alçada 4 (escalat uniformement) i que estarà situat amb el centre de la base de la seva capsula contenedora al punt `posPorter` (posició ja inicialitzada en el codi) i mirant en direcció X-. Observació: el model `legoman.obj` inicialment mira en direcció Z+.

També hauràs d'utilitzar el model del cub per a construir les tres parets del camp de joc: `Paret1`, `Paret2` i `Paret3`. Aquestes parets envolten el camp de joc en les tres vores on no es troba el legoman. La `Paret1` ha de tenir mides 20x2x0.2 (mida en X, Y i Z respectivament) i el centre de la seva base al punt (0,0,-6.9). La `Paret2` ha de tenir mides 20x2x0.2 i el centre de la seva base al punt (0,0,6.9). I la `Paret3` ha de tenir mides 0.2x2x14 i el centre de la seva base al punt (-9.9,0,0).

2. (2 punts) L'escena s'ha de poder visualitzar amb una càmera en tercera persona que permeti inicialment 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. 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  $\psi = M\_PI/3.0$  i  $\theta = 0$ .

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

3. (1.5 punts) 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 punt) Afegeix la possibilitat que prement la tecla 'F' el focus de llum passi a ser un focus d'escena a la posició (0,10,0) de l'escena. Tornant a prémer la tecla 'F' el focus serà de nou el focus de càmera descrit a l'exercici anterior.

5. (2 punts) Per a poder jugar cal que feu el següent:

- (a) Implementa el mètode `rebotaParets()` i fes que modifiqui la direcció de la pilota (representada a la variable `dirPilota`) de manera que aquesta reboti en les parets del camp. Fixa't que el rebot en cada paret implica únicament el canvi de signe en una de les components (X o Z) de la direcció de la pilota.

- (b) Afegeix la possibilitat que el legoman (porter) es mogui en la direcció de l'eix Z, entre els punts `pmin=(9,0,-5)` i `pmax=(9,0,5)` (el centre de la base no pot sortir-se del rang definit per aquests dos punts). 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- (corresponent a esquerra i dreta vist des del punt de vista del legoman). Pots fer que cada cop que es pitja la tecla el legoman es desplaci 0.5 unitats en la direcció indicada.

- (c) Els mètodes `rebotaPorter()` i `canviaDirecció()` els teniu ja implementats. Aquests mètodes fan que la pilota reboti en el porter (si aquest es troba en una posició en què toca la pilota).

Sí que cal, però, que complementis el mètode `tractamentGol()` per a què en cas de gol, la pilota desaparegui (no es pinti) i es compti el gol. Quan l'usuari torna la pilota a l'inici (amb la tecla `Key_I`) la pilota ha de tornar a aparèixer.

- (d) Afegeix un element d'interfície adient per a visualitzar el nombre de gols que s'han comptabilitzat fins al moment.

6. (2 punts) Afegeix una segona càmera que serà una càmera de vista en planta. Aquesta càmera s'ha de situar a una altura de 18 sobre el terra i ha de mirar cap al centre de l'escena. El legoman (porter) ha de quedar mirant cap a la part de dalt del viewport (veure imatge a `EscSol16.png`). 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 tota l'escena. La càmera no es modificarà amb interacció del ratolí i no deformarà l'escena en cas de redimensionament del viewport.

Cal afegir un element d'interfície adient per a poder decidir en quina de les dues càmeres (la càmera inicial en 3ª persona o la de vista en planta) es vol veure l'escena.