

Cal que llegiu detingudament les instruccions i tot l'enunciat abans de començar a fer res!

Instruccions

1. Disposes d'un arxiu pdf a `~/examen/assig/idi/doc/LabIDI_Index.pdf` que conté tota la informació de les transparències de laboratori indexades de manera que siguin fàcils de trobar.
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-2122Q1` 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, NO pots modificar la classe `ExamGLWidget`! Si ho fas es penalitzarà.**
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-2122Q1`):

```
make distclean
tar zcvf pompeu.fabra.tgz *
```

ALERTA: si el que segueix a `zcvf` és el nom d'un arxiu que existeix, es sobreescriurà!

És important el `'make distclean'` per 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-2122Q1` 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 l'arxiu a <https://examens.fib.upc.edu>

Nota: Obrint `~/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 terra de 30×24 ubicat en el pla $Y=0$ i amb el seu vèrtex de coordenades mínimes a l'origen, un avió d'alçada 1 amb el centre de la base de la seva capsa contenidora al punt $(2, 0, 2)$ i un Patricio d'alçada 1 amb el centre de la base de la seva capsa contenidora a l'origen. El Patricio i l'avió miren cap a les $Z+$. A l'escena també hi ha un hangar que no hauràs de modificar.

Els paràmetres inicials de la càmera no estan ben calculats, la `viewMatrix` està mal inicialitzada i només es pot modificar interactivament l'angle ψ . La imatge de l'arxiu `EscIni.png` mostra la visualització inicial.

Hi ha un mètode `creaBuffers` per a cada model (Terra, Patricio, Avió i Hangar). Aquests mètodes tenen 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 Fragment Shader. **Observació: Analitza el codi donat abans d'implementar els exercicis demanats.**

En la valoració de l'exercici 6 tindrà molta importància el disseny i la usabilitat de la interfície.

Una solució a tots els exercicis excepte el 6 la pots trobar a `~/examen/assig/idi/examen/examen`.

1. (1 punt) Modifica l'escena per a que l'avió tingui alçada 1.5 (escalat uniformement) i estigui situat inicialment amb el centre de la base de la seva capsula contenidora al punt (25, 0, 12). El Patricio ha de tenir alçada 2 (també escalat uniformement) i estar situat amb el centre de la base de la seva capsula contenidora al punt (15, 0, 12) i mirant cap a X- (mirant en direcció a l'hangar).
2. (1.5 punts) Implementa correctament el càlcul de la càmera (**viewMatrix** i **projectMatrix**) per tenir una càmera perspectiva en 3a persona de manera que l'escena es vegi en tot moment sencera, centrada, sense deformació i ocupant el màxim del viewport, encara que l'usuari modifiqui la finestra gràfica. Considera que l'altura de l'hangar (altura màxima de l'escena) és de 2.5.

Cal afegir que es pugui fer també rotació de l'angle θ , de manera que quan l'usuari mou el ratolí de baix cap a dalt en el viewport, l'angle θ es decrementa i per tant la càmera es mou cap avall respecte l'escena. Quan l'usuari mou el ratolí de dalt cap a baix en el viewport, l'angle θ s'incrementa. Els valors inicials dels angles han de ser: $\psi = 0.5$ i $\theta = -0.5$ (en radians). Una imatge de la solució als exercicis 1 i 2 la pots veure a l'arxiu `EscEx1-2.png`.

3. (2 punts) Modifica adientment el **Vertex Shader** i el **Fragment Shader** per afegir a l'escena el càlcul d'il·luminació usant el model d'il·luminació de Phong calculat al **Fragment Shader**. El focus de llum és inicialment blanc i de càmera, situat exactament a la posició de la càmera. La posició del focus es passa com a uniform als shaders.

Fes que prement la tecla F el focus canviï a un focus de model situat al centre de la cara posterior de la capsula contenidora de l'avió (a la cua de l'avió). El primer cop que es prem la tecla 'F' el focus passa a ser de model, i si es torna a prémer la tecla F, el focus torna a ser l'inicial de càmera. El comportament de la tecla 'F' és cíclic.

Modifica també el material del terra (mètode `iniMaterialTerra()`) per a què passi a ser de color cian i brillant.

4. (1.5 punts) Afegeix una segona càmera perspectiva, Càmera-2, que estigui situada a la posició del Patricio (posició X y Z del centre de la capsula contenidora) però a alçada 3 damunt del terra. Aquesta càmera mirarà en direcció paral·lela a l'eix X en sentit negatiu, de manera que mira cap a on mira el Patricio. L'angle d'obertura d'aquesta càmera serà fix de 90 graus, i els plans de retallat han de permetre veure tot el que està davant del Patricio. Aquesta càmera no permet interacció amb el ratolí. Mitjançant l'ús de la tecla C l'usuari ha de poder canviar de la Càmera-1 a la Càmera-2 i a la inversa.

5. (1.5 punts) Implementa el moviment de l'avió:

- Utilitza la tecla `key_Right` (fletxa dreta) per fer que l'avió es mogui circularment al voltant del Patricio (centre de l'escena) i orientat com si es mogués endavant. És a dir, si miréssim l'escena des de dalt l'avió giraria al voltant del Patricio en sentit horari. Fes que l'avió doni la volta completa al Patricio pitjant 32 cops la tecla `key_Right`.
- Utilitza la tecla `key_Up` (fletxa amunt) per fer que l'avió pugi una unitat damunt del terra (desplaçament una unitat en sentit Y+). El valor màxim d'alçada al que pot estar la base de la capsula de l'avió és 5.
- Utilitza la tecla `key_Down` (fletxa avall) per fer que l'avió baixi una unitat (desplaçament una unitat en sentit Y-). El valor mínim d'alçada al que pot estar la base de la capsula de l'avió és 0 (alçada inicial).

6. (1.5 punts) Afegeix a la interfície:

- a) Un element d'interfície que permeti decidir quina càmera volem tenir activada en cada moment. Aquest element ha d'estar coordinat amb l'efecte de la tecla C.
- b) Un element d'interfície que permeti controlar l'alçada de l'avió al damunt del terra. Aquest element s'ha d'actualitzar correctament amb l'efecte de les tecles `key_Up` i `key_Down`.

7. (1 punt) Afegeix la possibilitat de fer un "reset" (reinici) de tot el comportament de l'aplicació:

- Deixar l'avió a la posició inicial (posició descrita a l'exercici 1).
- Posar activa la Càmera-1 (càmera inicial) amb els valors inicials de la càmera en tercera persona de l'exercici 2.
- Posar el focus com a focus de càmera.

Aquest "reset" es farà mitjançant la tecla R i ha de reiniciar també els elements d'interfície adientment.