

# Interacció i disseny d'interfícies

## Exercicis tipus examen d'OpenGL

Dept. de Ciències de la Computació

Professors IDI

### 1 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 correu el risc que es pugui considerar el vostre examen com una còpia.
2. Partirem del codi que teniu a `examen.tgz` (adjunt a aquesta tasca). 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 cal 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 `INDI_ex_opengl_DNI.tgz`, on substituireu `INDI` pel vostre nom d'usuari. Per exemple, l'estudiant amb dni `11111111H` (des d'una terminal en la que s'ha col·locat dins del directori `examen`):

```
make distclean
tar cvzf INDI_ex_opengl_11111111H.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 `INDI_ex_opengl_DNI.tgz` que és el que heu de lliurar. **Feu la comprovació**, desplegant aquest arxiu **en un directori completament buit**, que el codi que lliureu compila (fent `qmake; make`) i executa correctament.
6. Finalment, lliureu el fitxer a <https://atenea.upc.edu> mitjançant la tasca "Lliurament OpenGL".

## 2 Sessió: Exercicis tipus examen

L'objectiu d'aquesta última sessió abans de l'examen és que us familiaritzeu amb el tipus d'exercicis que poden sortir a l'examen, i que pogueu practicar en les mateixes condicions que tindreu a l'examen: partir d'un codi ja implementat, haver d'entendre el que fa i com ho fa, i resoldre els exercicis que demana l'enunciat (que poden ser entre cinc i set, depenent de la dificultat dels exercicis).

En aquesta llista força àmplia d'exercicis, us trobareu alguns exercicis que són incompatibles amb els anteriors. Aquests exercicis estan marcats com a ► **Nova escena**:. Quan us trobeu aquest exercici guardeu l'exercici anterior i feu-ne un de nou (que pot ser a partir del que ja teniu) amb les noves demandes. Això no us passarà el dia de l'examen, **a l'examen tots els exercicis es podran implementar en el mateix codi, no hi haurà incompatibilitats entre dos exercicis del mateix examen**.

### 2.1 Exercicis

1. Afegeix a l'escena un segon Patricio, de la mateixa mida, que estigui situat al damunt del que ja tenim però cap per avall, és a dir els dos caps dels dos patricios tocant-se i tots dos mirant cap al mateix costat.

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  $\psi$  i  $\theta$ ), és a dir, l'usuari ha de poder modificar aquests angles utilitzant el ratolí.

2. Afegeix a l'escena el càlcul d'il·luminació usant el model de Phong de manera que el càlcul es faci al Vertex Shader i a l'escena hi hagi un focus de llum de càmera (exactament a la posició de la càmera) de color cian.
3. Afegeix al codi que amb el botó dret del ratolí i amb un moviment vertical es pugui fer un zoom modificant l'angle d'obertura de la càmera.
4. Implementa la funcionalitat anterior (Zoom) també usant un Slider afegit a la interfície. L'Slider ha de permetre modificar l'angle d'obertura de la càmera dins del rang de valors que aquest angle permet.

Si l'usuari modifica el valor de l'angle d'obertura mitjançant el botó dret del ratolí sobre la finestra OpenGL (usant la funcionalitat de l'exercici anterior), aquest nou valor de l'angle també s'ha de veure reflectit en l'Slider de la interfície.










5. ► **Nova escena**: Modifica l'escena i treu els dos Patricios per a pintar una vaca (model `cow.obj`) d'alçada 0.5 i posicionada amb el centre de la cara de la capsa on té les potes al punt (1, -1, 0), és a dir, la vaca ha d'estar dreta sobre el terra. L'escena també tindrà

un Patricio d'alçada 0.25 situat damunt de la vaca, de manera que el centre de la base de la seva capsa estarà en el punt (1, -0.5, 0).

6. Comptant amb l'escena de l'exercici anterior, fes que quan l'usuari prem la tecla , tots dos objectes (la vaca i el Patricio) giren conjuntament al voltant de l'eix Y 30° ( $M\_PI/6.0$  radians). Fes que l'angle de gir s'incrementi en 30° cada cop que es prem la tecla .
7. Modifica la càmera per a tenir una càmera perspectiva situada a una cantonada de la paret, a la posició (-1, 1, -1), mirant cap al centre del Patricio que es troba damunt la vaca, amb un angle d'obertura de 60° ( $M\_PI/3.0$  radians) i Znear i Zfar adjacents per a què sempre es vegi l'escena que queda davant de l'observador. La càmera sempre mirarà cap al centre del Patricio quan aquest es mou (juntament amb la vaca). No pot haver deformacions de l'escena si l'usuari fa un resize.
8. Fes que la vaca, que –com potser hauràs comprovat– no té informació de material, es pinti d'un mateix material gris brillant. Aquest material el pots definir directament al Vertex Shader. Fixa't que et caldrà alguna manera de dir-li al Vertex Shader quan se li està enviant a pintar la vaca (i ha d'utilitzar aquest material) i quan no.
9. Fes que en prémer la tecla  el Fragment Shader pinti tota l'escena fent franges blanques o negres depenent de l'altura (coordenada Y) del fragment. Les franges han de ser de 10 píxels de gruix. Si es torna a prémer la tecla  es torna a la il·luminació original. Per fer aquest exercici, recordeu que la variable `gl_FragCoord` us dona les coordenades del fragment i és de tipus `vec4`. El Vertex Shader no cal que el toqueu en aquest exercici.
10. Fes que en prémer la tecla  el focus passi a ser d'escena, a la posició (1,1,1) de l'escena (SCA) i de color blanc. 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 (de càmera i de color cian). Si es torna a prémer  s'ha de tornar a pintar tot amb el color del focus original i així successivament.
11. Implementa la funcionalitat anterior també usant un Radio Button afegit a la interfície. El Radio Button tindrà dos valors possibles amb text: "Focus escena" i "Focus càmera". Si està activat "Focus escena" el focus ha de ser el d'escena a la posició (1,1,1), mentre que si està activat "Focus càmera" el focus ha de ser el de càmera (a la posició (0,0,0)) i de color cian. Si l'usuari canvia aquest focus mitjançant la tecla  s'ha de veure reflectit en el Radio Button de la interfície. Inicialment s'ha de tenir seleccionat "Focus Escena" i per tant el focus inicial ha de ser el d'escena.
12. ► **Nova escena:** Modifica l'escena per a que, en lloc del Patricio inicial, hi hagi 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 (-2, -1, 0) i estarà mirant en direcció X+, el segon Patricio serà d'alçada 2.5 (també escalat uniformement) i ha de tenir el centre de la seva base al punt (2, -1, 1) i estarà mirant en direcció X-.

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  $\psi$  i  $\theta$ ), és a dir, l'usuari ha de poder modificar aquests angles utilitzant el ratolí.

13. Fes que usant les tecles  i  el primer Patricio (el petit) es mogui sobre la pista en una direcció paral·lela a l'eix X, de manera que si l'usuari prem la tecla  el Patricio es mou en direcció X+ mentre que si l'usuari prem la tecla  el Patricio es mou en direcció X-.
14. Posiciona un focus de llum blanca al damunt del Patricio petit. Aquest focus de llum s'ha de moure juntament amb el Patricio sobre l'eix de les X quan aquest es mou per efecte de les tecles de l'exercici anterior.
15. Fes que en prémer la tecla  es canviï de càmera de manera que ara passem a tenir una càmera just a sobre del primer Patricio (el petit), a alçada 1.5 i mirant cap al centre del segon Patricio (el gran). Aquesta segona càmera ha de ser una càmera perspectiva amb angle d'obertura de  $M_{PI}/2.0$  radians (90 graus), posició sempre sobre el Patricio petit (també quan es mou amb les tecles  i ) i Znear i Zfar adjacents per a veure tot el que es pugui veure de l'escena des de la seva posició i orientació. La càmera no hauria de deformar l'escena en cas de redimensionament del viewport.  
Prement de nou la tecla  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.
16. Implementa la funcionalitat anterior també usant un checkbox afegit a la interfície, que tingui com a text "Càmera Patricio" i que si està activat la càmera ha de ser la del damunt del Patricio, mentre que si no està activat la càmera ha de ser la càmera que veu tota l'escena.  
Si l'usuari canvia aquesta càmera mitjançant la tecla  s'ha de veure reflectit en el checkbox de la interfície. Inicialment el checkbox no ha d'estar activat perquè la càmera inicial no és la del Patricio.
17. Afegeix a la interfície tres Sliders amb valors entre 0 i 100 que permetin decidir els tres components del color (R, G i B) del terra i la paret. Es modificarà la component difusa del material. El valor 0 de l'Slider correspondrà amb el valor 0 de la component de color i el valor 100 de l'Slider amb el valor 1 de la component. Cal que el canvi de color del terra+paret sigui interactiu quan es mouen els Sliders.