

Interacció i Disseny d'Interfícies: Activitat 3

24 de maig de 2022

Instruccions

1. Has de partir del codi que tens a `activitat3.tgz` (el podeu trobar en el Campus digital). Has de desplegar aquest arxiu en un directori on disposaràs tots els fitxers amb els que has de treballar.
2. Els exercicis que es demanen només requereixen canvis a la classe *MyGLWidget* (.cpp i .h), als *shaders* i al fitxer *MyForm.ui* usant el designer. No has de modificar cap altre fitxer dels que se't proporcionen.
3. El codi que lliuris ha de compilar i executar correctament. Si no compila o dóna error d'execució, l'avaluació de l'exercici serà un 0, sense excepció.
4. Per a fer el lliurament has de generar un arxiu TGZ que inclogui tot el codi del teu exercici i que es digui `activitat3_NIF.tgz`, on substituiràs NIF pel teu número de NIF amb la lletra inclosa.

Per exemple, l'estudiant amb NIF 12345678Z (des d'un terminal en el que s'ha col·locat dins del directori `activitat3`) farà:

```
make distclean
tar zcvf activitat3_12345678Z.tgz *
```

És important fer el `'make distclean'` per a esborrar els arxius binaris generats; que el DNI sigui el correcte (el teu); i que hi hagi el sufix `.tgz`.

5. Has de lliurar l'exercici usant la tasca corresponent del Campus digital abans del **dilluns 5 de juny de 2023** a les 23:55.

Enunciat

El codi esquelet proporcionat dibuixa d'una escena composta de diferents elements(Figura 1):

- Un background, format per les parets, el terra, la taula i altres elements decoratius.
- dues espelmes (tenen el mateix model)
- dos snitch, que son les dues pilotes alades que estan a la base de les espelmes (també comparteixen el mateix model)
- les finestres i els flascons, que estan per separat doncs són tots elements de vidre i voldrem que siguin translúcids.

Es donen ja implementats els mètodes que construeixen els VAOs i VBOs dels models (`creaBuffersXXXXXX()`) i els mètodes que creen les seves transformacions.

Les tecles `W` / `S` permeten moure l'espelma activa en vertical, i `A` / `D` el horitzontal. Podeu alternar l'espelma activa que es modifica amb la tecla `T`

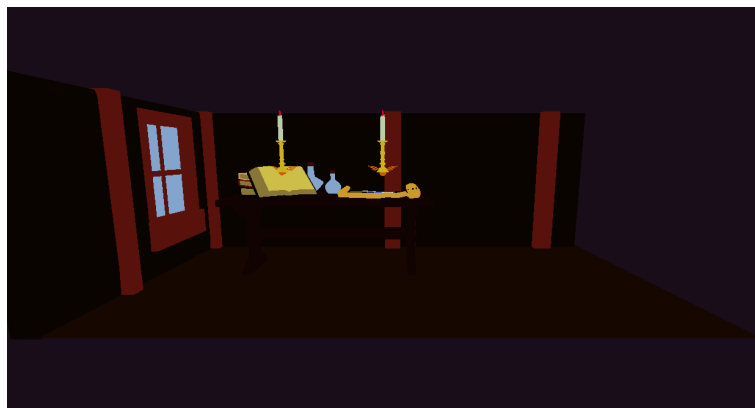


Figura 1: Escena inicial.

La càmera està calculada de manera arbitrària, de forma que permet veure el conjunt. També es dona implementat el gir de la càmera per coordenades euler mitjançant el mouse.

Per a resoldre aquesta activitat es demana el següent:

1. (0.25 punts) Modifica el fragment shader de forma que la il·luminació de l'escena sigui estrictament la component ambient. El color de la llum ambient és (0.1,0.1,0.1) i s'ha de passar com a `uniform` des del codi.



Figura 2: Escena amb il·luminació ambient. Que ens tornin els diners!

2. (2 punts) Completa el model de Phong al **Fragment Shader** afegint el component difús i l'especular. Hi haurà un focus d'escena de llum blanca tènue (0.6,0.6,0.6) situat a la posició (0, 5, 0) en el sistema de coordenades de l'escena (SCA o world coordinates). Cal passar les coordenades de la llum i el color de la llum com a uniforms. L'efecte aconseguit per la il·luminació es mostra la figura 3.

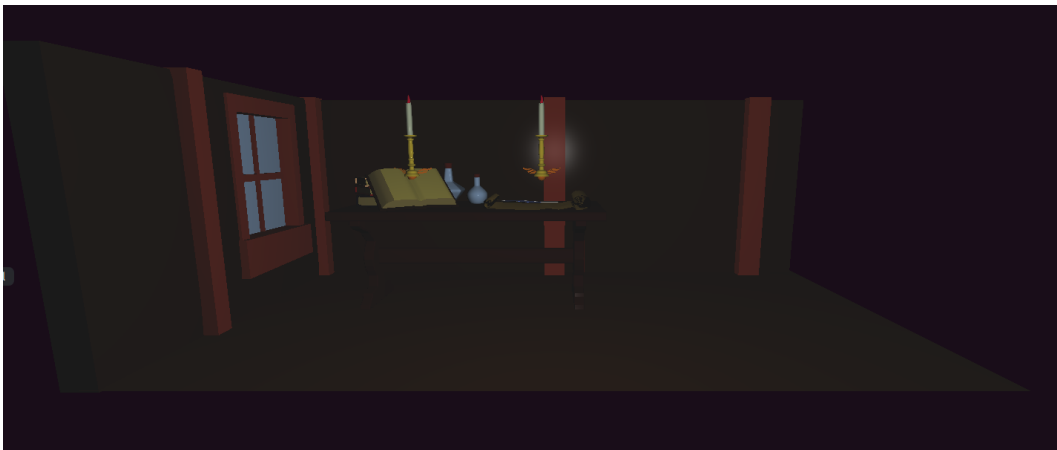


Figura 3: Escena amb il·luminació de Phong

3. (1 punts) Creeu un element d'interfície adequat per a seleccionar quina és l'espelma activa. L'element s'haurà d'actualitzar en el cas de canviar l'espelma activa amb la tecla **T**.
4. (2 punts) Volem que els snitch orbitin cadascun al voltant de la seva espelma en el pla horitzontal. L'orbita tindrà un radi d'1 des del centre de l'espelma. La tecla **R** modifica l'angle de rotació de l'snitch associat a l'espelma actualment seleccionada. És a dir, que quan canviem d'espelma seleccionada, la R passa a afectar a l'altre l'snitch. Per ajudar-vos, l'array `AngleSnitch[indexEspelmaActiva]` us permet saber quin és l'angle de rotació que cal aplicar a cada snitch. Tingueu present que si l'espelma es mou, l'snitch ha d'acompanyar-la.

Us farà falta modificar el mètode `modelTransformSnitch(int i)`, que rep com a paràmetre l'índex de l'snitch que volem col·locar. L'alçada dels

snitch ha de ser de 0.75, i ha d'estar col·locat aproximadament a l'alçada de la flama de l'espelma.

Podeu buidar la funció `modelTransformSnitch(int i)` i crear el vostre codi, que pot inspirar-se en el del mètode `modelTransformEspelma(int i)`

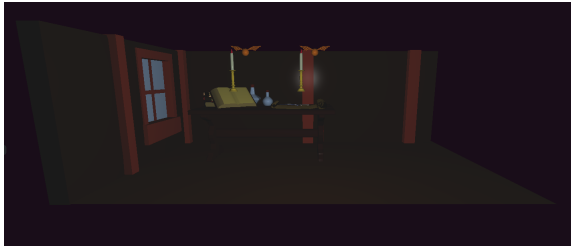


Figura 4: Posició inicial dels snitch's

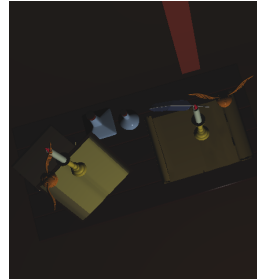


Figura 5: Rotem amb R

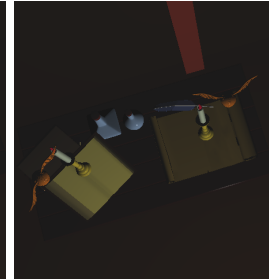


Figura 6: més rotat...

5. (1 punts) Creeu un element d'interfície gràfica adequat per controlar la rotació de l'snitch actual. Mantingueu la sincronia:

- amb els rotacions festes amb la tecla **R**.
- quan canviem d'espelma/snitch actual **T** l'element ha de passar a mostrar la rotació que té actualment l'espelma seleccionada.

6. (2 punts) Ara farem que els snitch es comportin com a focus de llum. Per tant, en la posició de cadascun d'ells haurem de simular un focus de color (0.5,0.5,0.5). Evidentment, quan l'espelma es mogui o el snitch roti, la llum es mourà exactament igual que l'snitch.

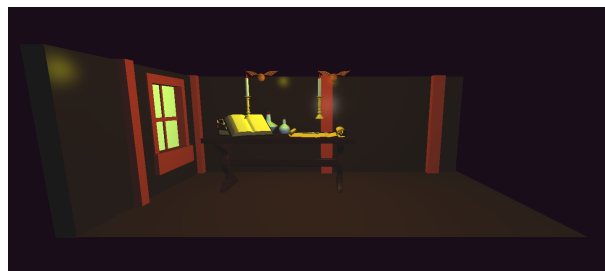


Figura 7: Focus als snitch

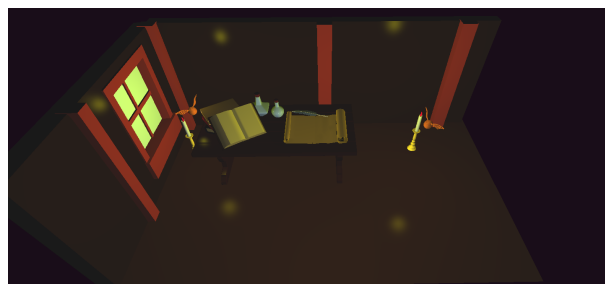


Figura 8: Jugant amb els snitch i les espelmes

7. (1 punt) Modifiqueu la interfície gràfica per tal que es pugui apagar/encendre la llum actualment seleccionada.
8. (0.75 punts) Ara farem que els objectes de vidre (finestres i flascons) siguin translúcids. A tal efecte ens cal:
- activar el mode de blending a openGL amb `glEnable(GL_BLEND);` dins de `initializeGL()`.
 - Renderitzar primer els objectes sòlids i després els objectes translúcids. La idea és que quan es dibuixa el translúcid sobre el sòlid, es barregen els colors.
 - La barreja s'ha de fer usant el canal alfa(a) dels colors: (r, g, b, \mathbf{a}) . Al fragment shader forçarem una $a = 0.5$ per l'objecte translúcid (finestres) i una $a = 1$ pels l'objectes sòlids (la resta). El truc és usar un uniform per informar al shader de si ha de posar la component alpha del color resultat a $a = 1$ (opac) o $a = 0.5$ (translúcid) al color final que ha calculat el shader.
 - Al `paintGL()`, abans de pintar l'objecte sòlid, activem el mode de barreja per defecte: `glBlendFunc(GL_ONE, GL_ZERO);`
El que fa és simplement pintar l'objecte de forma normal.
 - Al `paintGL()`, un cop pintat l'objecte sòlid, abans de pintar un objecte translúcid (que té una alfa(a) menor a 1), hem d'activar el mode de barreja:
`glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);`
Això fa que el color final sigui la barreja del translúcid i del sòlid ja pintat anteriorment en el framebuffer.

La figura 9 mostra el resultat esperat.

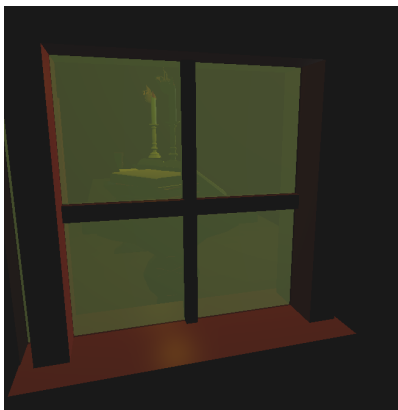


Figura 9: Mirem per la finestra!

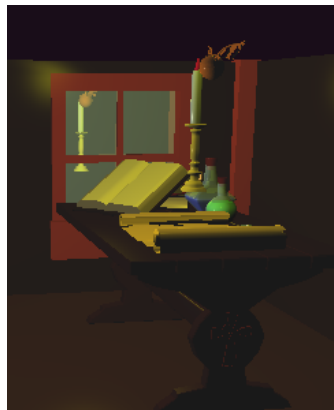


Figura 10: Fem fora l'espelma

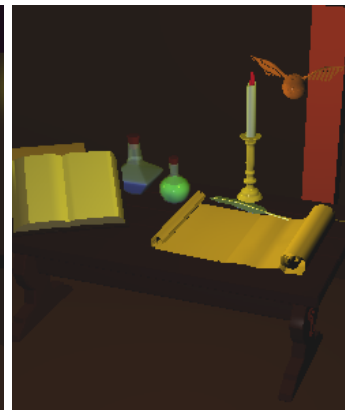


Figura 11: Flascons transparents.

* * *

9. (1 punts / **EXTRA!**) Finalment, ens agradaria crear un efecte especial sobre el terra per simular textura de rajoles. En comptes d'usar les normals de l'objecte en els càlculs d'il·luminació, les generarem nosaltres a partir d'una model de perfil de rajola simplificat.

Només aplicarem aquesta manipulació de les normals al fragments que corresponen al terra. I com sabem que estem dibuixant el terra? Doncs quan detectem que la coordenada y del vèrtex en coordenades d'escena o aplicació (SCA) sigui més petita que 0.1. Ja veieu que ens farà falta passar les coordenades SCA del vèrtex al fragment shader.

A la Figura 12 teniu l'esquema que heu de seguir per calcular les normals simulades. Definim que cada rajola té amplada $W=2$ i la rampa que separa les rajoles és d'amplada $d=0.2$. Només cal que useu la coordenada x en SCA, obteniu el residu de dividir per W i determineu a quina regió esteu (pujada, pla o baixada), i canvieu la normal en conseqüència usant els valors de la figura.

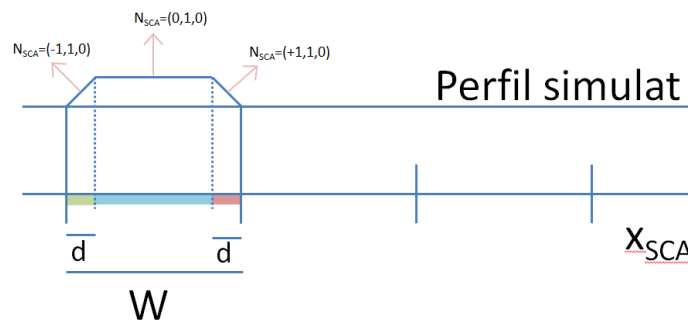


Figura 12: Esquema del càlcul de les normals per simular rajoles.

Fent exactament el mateix amb la coordenada z en SCA ja tindreu una quadrícula. Recordeu adaptar les normals a l'eix z .

IMPORTANT: Les normals que obtenim estan en SCA, però la il·luminació treballa amb SCO. Per tant, caldrà que la passeu els vectors normals a coordenades SCO multiplicant per la VM. El resultat hauria de ser semblant al de la figura 13.

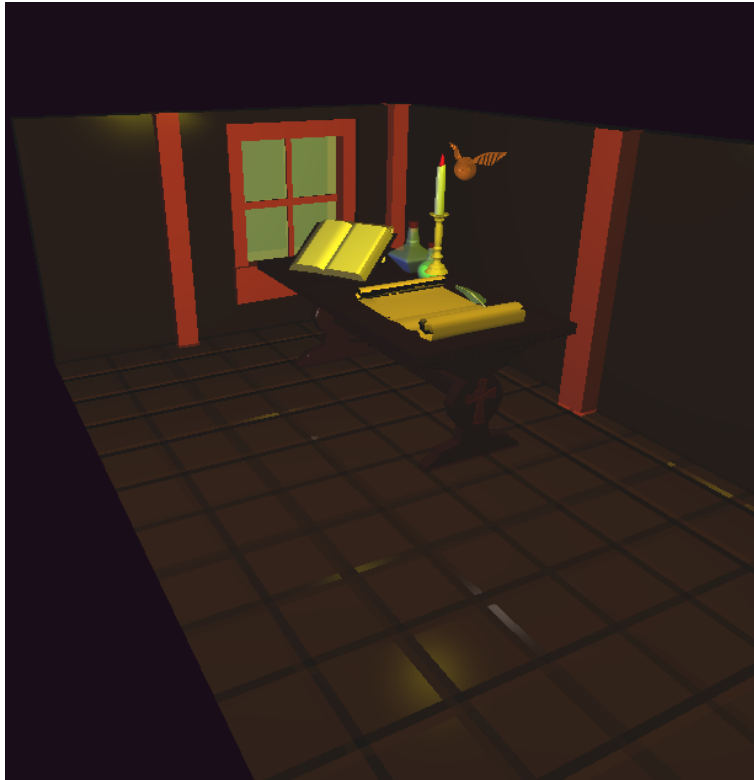


Figura 13: El terra enrajolat.

10. (0.25 punts / **EXTRA!**) Afegiu una mica de soroll al shader anterior per donar un efecte d'irregularitat al terra. És molt senzill, crideu a la funció `noise` passant les coordenades x i y en SCA, ens retornarà un valor aleatori r . Modifiqueu la normal obtinguda a l'exercici anterior sumant-li $r * \text{vec4}(1, 0, 0, 0)$ (feu la suma en SCA)

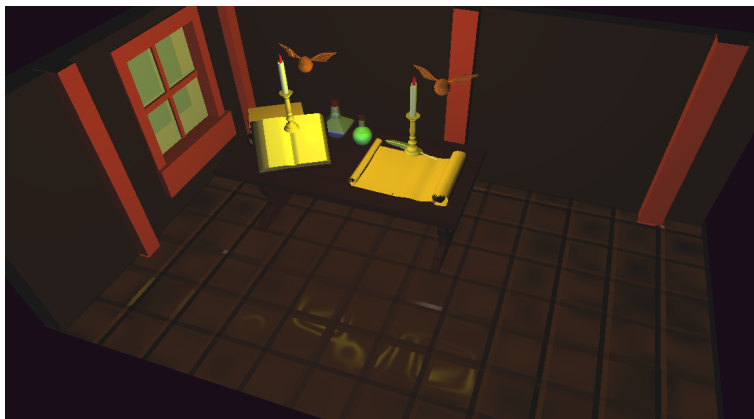


Figura 14: El terra amb soroll.