Interacció i Disseny d'Interfícies: Exercici 1

Jordi Esteve

11 de març de 2019

Instruccions

1. Aquests exercicis són individuals, així que només pots entregar codi que hagis generat tu; no pots fer servir codi que altres estudiants hagin compartit amb tu (ni que tu hagis compartit amb d'altres estudiants). Altrament es considerarà còpia.

2. Has de partir del codi que tens a `exercici1.tgz` (el podeu trobar en el Campus digital). Has de descomprimir aquest arxiu en un directori teu. Es crearà un subdirectori `exercic1` on tindràs tots els fitxers amb els que has de treballar.

3. Els exercicis que es demanen només requereixen canvis a la classe *MyGLWidget* i als shaders. No has de modificar cap altre fitxer dels que se't proporcionen, ni tampoc canviar el seu nom. A més els mètodes `initializeGL` i `pintaArbre` de la classe *MyGLWidget* no els pots modificar.

4. El codi que lliuris ha de compilar i executar correctament en els ordinadors del laboratori. Si no compila o dóna error d'execució, l'avaluació de l'exercici serà un 0, sense excepció.

5. Per a fer el lliurament has de generar un arxiu TGZ que inclogui tot el codi del teu exercici, sense fitxers binaris i que es digui `INDI_exercici1_DNI.tgz`, on substituiràs `DNI` pel teu DNI amb lletra majúscula i sense guions.

   Per exemple, l'estudiant amb DNI 12345678A (des d'una terminal en la que s'ha situat dins del directori exercici1) farà:

   ```
   make distclean
   tar zcvf INDI_exercici1_12345678A.tgz *
   ```

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

6. Has de lliurar l'exercici a l'activitat del Campus digital) abans del **dimarts 19 de març de 2018** a les 23:55.

Enunciat

El codi que proporcionem ofereix el mètode `pintaArbre()` que pinta un arbre centrat en el punt (-0.5, 0, 0) els vèrtexs del qual ja estan inicialitzats en el mètode `creaBuffersArbre()`. Aquest arbre es pinta inicialment de color negre, color indicat directament en el fragment shader (vegeu figura 1). Es demana que, donat el codi que us hem proporcionat, resolgueu els següents exercicis:

1. *(3 punts)* Afegeix color a l'arbre, fent que aquest color sigui un nou atribut del vèrtex i afegint la creació del VBO corresponent als colors, l'enviament d'aquest a la tarja gràfica i tot allò que cal en els dos shaders (basicShader.vert i basicShader.frag) per a què l'arbre es vegi pintat en colors. Els colors han de ser el marró (RGB = (1.0, 0.6, 0.3)) per als 6 primers vèrtexs (tronc) i el verd per als sis últims vèrtexs (fulles).

2. *(4 punts)* Implementa el mètode
`void modelTransformArbre(glm::vec3 posicio, float escalat, float anglegir)`
fent que calculi i enviï a la tarja gràfica una transformació geomètrica que escali l'arbre amb el factor `escalat`, faci una rotació d'un angle d'`anglegir` radians al voltant d'un eix que passa pel centre de l'arbre i és paral·lel a l'eix Y i posicioni el seu centre en el punt `posicio`. Ajusta el mètode `paintGL()` perqué pinti dos arbres: el primer el doble de gran, girat 0 graus i centrat a (-0.5, 0.0, 0.0), el segon la meitat de petit, girat 0 graus i centrat a (0.5, 0.0, 0.0).

3. *(3 punts)* Afegeix el codi necessari al mètode `keyPressEvent` per a què cada cop que l'usuari prem la tecla 'R' s'incrementi en 5 graus l'angle de rotació de l'arbre que es troba a l'esquerra, si prem la tecla 'T' s'incrementi en 10 graus l'angle de rotació de l'arbre que es troba a la dreta i si prem la tecla 'I' els dos arbres es tornen a veure com a l'inici (amb angles de rotació de 0 graus).

Pots executar `/home/public/indi/bin/exercici1` per provar la solució que es demana, prova prèmer les tecles 'R', 'T' i 'I' vàries vegades. La figura 2 mostra com ha de quedar la imatge (els angles de rotació dels dos arbres són 0º en aquesta imatge).

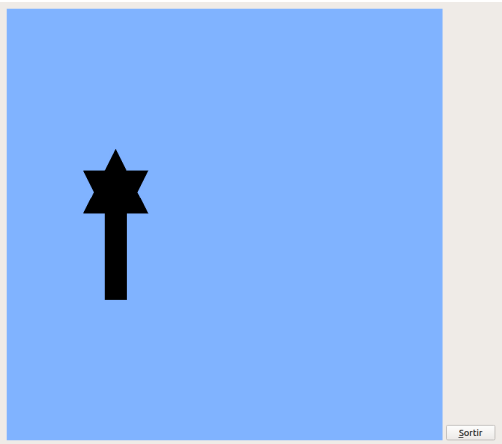

Figura 1: Escena inicial

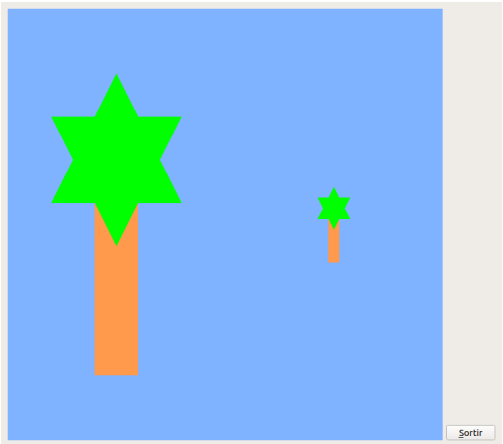

Figura 2: Escena final amb angles 0º