Interacció i Disseny d'Interfícies: Activitat 1

18 de març de 2024

Instruccions

1. Aquesta activitat és individual, 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 `activitat1.tgz` (el podeu trobar en el Campus digital). Has de descomprimir aquest arxiu en un directori, dins hi trobaràs 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.

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'activitat serà un 0, **sense excepció**.

5. Per fer el lliurament has de generar un arxiu TGZ que inclogui tot el codi de la teva activitat, sense fitxers binaris i que s'anomeni : `INDI_activitat1_[cognom1]_[cognom2]_[nom].tgz`.

   Per exemple, l'estudiant Pere Sans Martínez (des d'una terminal en la que s'ha situat dins del directori activitat1) farà:

   ```
   make distclean
   tar zcvf INDI_activitat1_Sans_Martinez_Pere.tgz *
   ```

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

6. Has de lliurar l'activitat a Campus digital) abans del **dimecres 27 de març de 2024** a les 23:55.

Enunciat

Volem construir el nostre propi “London Eye”, però a l’estil INDI: a cop de shader i OpenGL !

Es proporciona un codi bàsic que mostra una escena inicial on només es mostra el pedestal de la base de la sínia. (Fig.1):

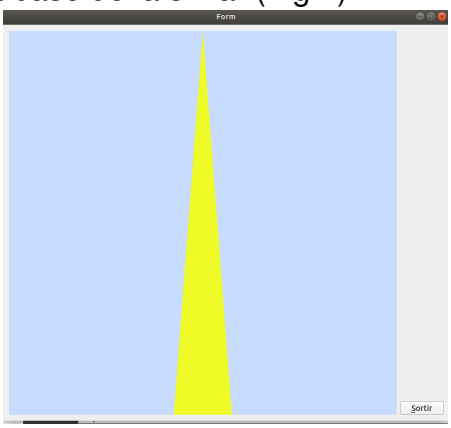


Figura 1: Els inicis sempre són durs...

A partir de l’esquelet de codi, resol els següents exercicis:

1. *(0.75 punts)* Modifiqueu el codi de `transformacioBase` de forma que el pedestal tingui una alçada total de 0.75 (manteniu l’amplada). Vegeu Fig. 2 pel resultat. **IMPORTANT**: no es poden tocar les coordenades del buffer.

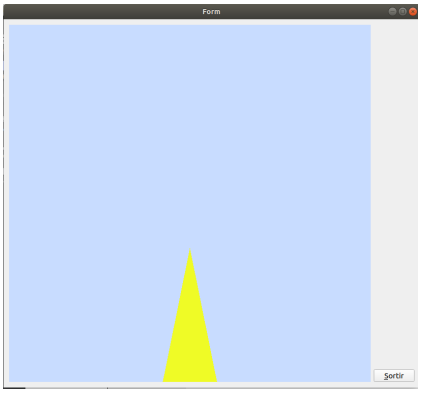


Figura 2: La base al seu lloc.

2. *(0.5 punts)* Descomenteu la crida a `pintaSector()` que trobareu dins de `paintGL()`. Això mostrarà un sector de la sínia. Modifica la creació del buffer del sector (sense modificar cap coordenada de posició!) per a què es mostrin els colors segons la Fig. 3.

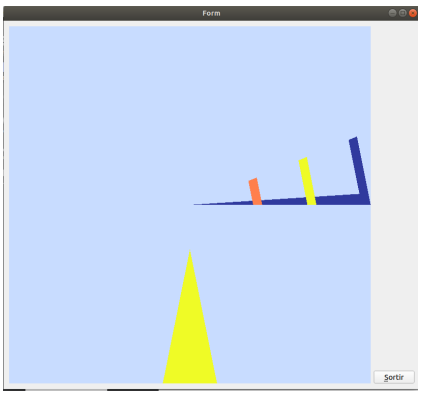
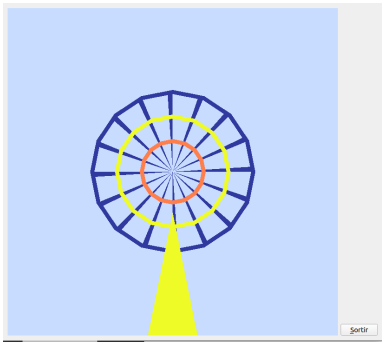


Figura 3: Sector amb colors

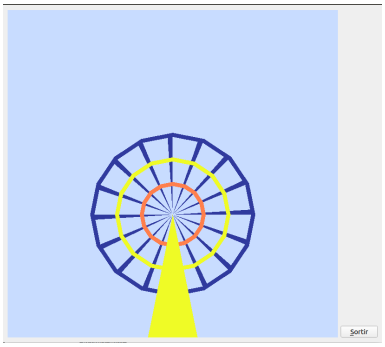
3. *(1.25 punts)* Dibuixeu el sector tantes vegades com sigui necessari fins a composar tota la sínia. Evidentment, cada sector s'ha de dibuixar aplicant una transformació diferent (vegeu Fig. 4, esquerra), després baixeu tots els segments de forma que el centre de la sínia estigui a (0,-25,0), us quedarà com la Fig. 4, dreta.

   IMPORTANT:

   - useu la variable `angleSectorNoria` per saber el nombre de repeticions,
   - recordeu que no podeu modificar les coordenades del buffer, cal usar transformacions.



(a) Sínia amb el centre a origen



(b) Sínia ben centrada.

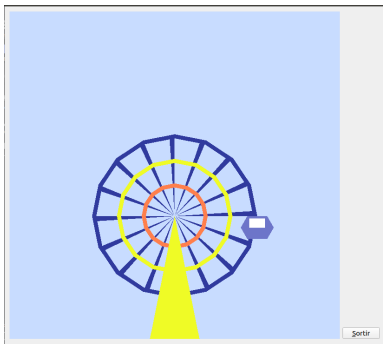
Figura 4: La sínia, WIP.

4. *(2 punts)* Feu que les tecles A i D facin girar la sínia sobre el seu eix, cadascuna en un sentit diferent. La base no es veurà afectada per la rotació.

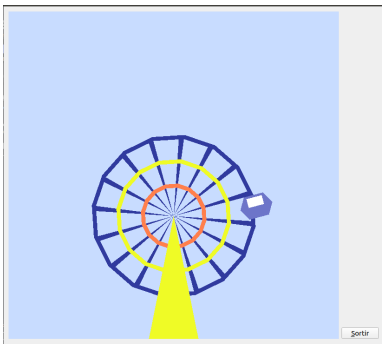
5. *(2 punts)* Descomenteu la crida a `pintaCistella()` que trobareu dins de `paintGL()`. Ara veureu la cistella que dona voltes a la sínia (mida King-size). Modifiqueu la funció `transformacioCistella()` per tal que:

   - Estigui escalada un factor 1/10 uniformement.

- El punt central de la seva part superior ha de quedar aliniada al punt on acaba el primer sector la sínia (el que queda horitzontal), és a dir, a la posició (x=0.5, y=-0.25).
- Feu que la rotació de la sínia també afecti a la cistella, vegeu Fig. 5.



(a) Cistella a la posició base.



(b) La cistella gira amb la sínia (!!!)

Figura 5: La sínia, WIP.

6. *(0.5 punts)* Arregleu la cistella, de forma que es mantingui sempre horitzontal. Vegeu Fig. 6.

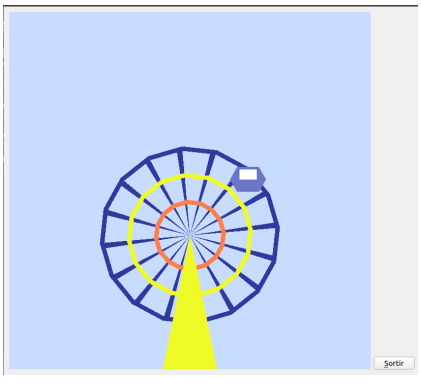


Figura 6: Hands up!

7. *(0.75 punts)* Modifiqueu el `fragment shader` per simular una reixa amb forats quadrats a la base de la sínia, tal i com mostra la Fig. 7. Caldrà que passeu una variable `uniform` per distingir quan estem renderitzant la base de la resta d'objectes de l'escena, i que useu `gl_FragCoord` i `discard`.

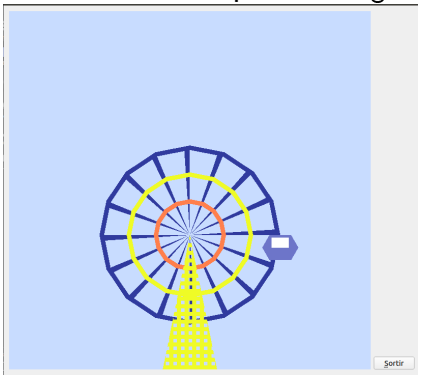


Figura 7: Base amb reixa.

8. *(1.25 punts)* Descomenteu la crida a `pintNuvol()` que trobareu dins de `paintGL()`. Apareixerà un rectangle blanc que ocupa tota l'escena. Convertirem aquest rectangle en un núvol gràcies al `fragment shader`. Com ho aconseguirem?

   - Al igual que hem fet amb la base, passarem una variable `uniform` per indicar quan estem pintant el núvol.
   - Haureu de passar al shader un array de cinc `vec3`:

     ```
     uniform vec3 nuvolPoints[5];
     ```

     L'array ja està definint a `MyGLWidget` i es diu `nuvolPos`. Es tracta de cinc punts que defineixen l'interior del núvol.
   - Al `fragment shader` descartarem aquells pixels que estiguin més lluny de 0.08 unitats de qualsevol dels punts de l'array `nuvolPos`. Necessitarem passar del `vertex shader` al `fragment shader` les coordenades dels vèrtex, per així poder-les consultar per cada píxel.

   Us haurà de quedar com mostra la Fig. 8.

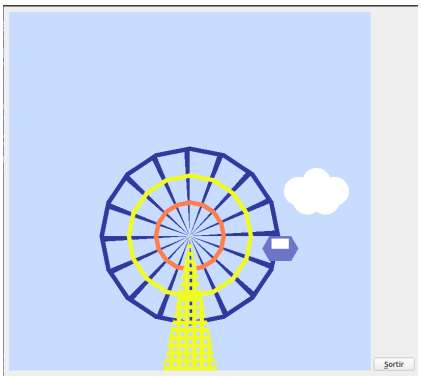


Figura 8: El núvol quadrat !!

9. *(1 punts)* Finalment, feu que les tecles A i D moguin també el núvol a esquerra i dreta respectivament.

10. **EXTRA** *(1 punt)* Poseu 8 cistelles equiespaiades en comptes d'una.