Laboratori IDI: OpenGL, bloc 2

Professors d'IDI, 2015-16.Q1

6 d'octubre de 2015

En aquest segon bloc, partirem d'un codi que pinta dos objectes, un polígon amb forma de caseta (amb un color diferent per a cada vèrtex) i un quadrat que fa de terra. Aquest codi el pots trobar a `/assig/idi/blocs/bloc-2`. La idea és que primer de tot et familiaritzis amb el codi que et passem i entenguis què fa i perquè cal cada línia del codi.

La durada d'aquest bloc és de 2 sessions de laboratori.

1 Sessió 4: Transformacions de càmera i càrrega d'OBJs

Un cop hem vist a classe la sessió 4 de laboratori, per a començar amb els exercicis, ► el primer que et caldrà fer és compilar i executar l'exemple que et passem per a veure què fa, i estudiar amb detall el codi a partir de l'explicació que s'ha fet a classe.

Quan ja hagis mirat i entès l'exemple d'aquest bloc, fes **tots** els següents exercicis en l'ordre en què estan, perquè defineixen una guia a seguir per a poder entendre tots els passos.

1.1 Exercicis:

1. ► Afegeix al codi de l'exemple un mètode `projectTransform ()` que implementi la crida a perspective amb paràmetres FOV = M_PI/2, ra = 1, znear = 1, zfar = 3, i enviï el uniform corresponent de la matriu de projecció al vertex shader, que farà el corresponent amb ella.

   És normal el que es veu considerant que la posició de la càmera és el punt (0,0,0) (matriu identitat en view transform)?

2. ► Afegeix al codi de l'exemple un mètode `viewTransform ()` que implementi la crida a lookAt amb paràmetres OBS = (0,0,2), VRP = (0,0,0), UP = (0,1,0), i enviï el uniform corresponent de la matriu view al vertex shader, que farà el corresponent amb ella.

   Un cop tenim tota la càmera perspectiva definida podem treure la rotació que té l'exemple en la matriu de transformació de model, que només pretenia que el terra es veiés amb la càmera per defecte.

3. ► Intercanvia l'ordre de pintat dels dos objectes (en el paintGL), és a dir, pinta primer la caseta i després el terra. Es continua veient bé?

   Activa el Z-buffer com s'ha explicat a classe i torna-ho a provar.

4. ► Modifica l'exercici traient la caseta i pintant en el seu lloc l'objecte definit pel fitxer `HomerProves.obj`. Carrega l'objecte (load) abans de construir els buffers i construeix dos VBOs per a aquest objecte, un a partir de les dades dels seus vèrtexs que el retorna el mètode `VBO_vertices()` de

la classe Model i l'altre a partir de la informació del seu material que ens retorna el mètode `VBO_matdiff()` i que podem usar com a informació de color per a passar-la al vertex shader.

5. ► Afegeix una interacció de teclat de manera que amb la tecla R el Homer roti cada vegada 45 graus (M_PI/4 radians) respecte l'eix vertical Y. Assegura't que el terra no rota, és a dir, hauràs de tenir una transformació de model diferent per al terra i per al Homer.

2 Sessió 5: Euler i objecte qualsevol

Un cop hem vist a classe la sessió 5 de laboratori, podeu continuar amb la llista d'exercicis següents. Us aconsellem que us guardeu el resultat de l'exercici de la sessió anterior.

2.1 Exercicis:

1. ► Afegeix a l'exercici que mostra el terra i el HomerProves la implementació necessària per a que, quan l'usuari fa una redimensió de la finestra, no hi hagi deformació de l'escena ni es retalli el que s'estava veient.

2. ► Ara volem visualitzar una instància del model `Patricio.obj`, però aquest model no està creat per a que els seus vèrtexs estiguin tots en el volum de visió que tenim definit fins ara. Has d'afegir al teu codi el càlcul de la capsa contenidora del model i pintar el Patricio centrat a l'origen (sense escalar). Com que no es vol que el Patricio s'escali per a què hi càpiga al volum de visió que tenim, caldrà que calculis també els paràmetres d'una càmera perspectiva que et permeti veure aquest objecte centrat a l'origen, sencer, i ocupant el màxim del viewport.

   Què ha passat amb el terra que teníem? Elimina (o comenta) el seu pintat al mètode paintGL.

3. ► Modifica el mètode `viewTransform ()` per a afegir el càlcul de la matriu *view* a partir de les transformacions mitjançant angles d'Euler. Inicialitza aquests paràmetres per a veure el que es veia a l'exercici anterior.

4. ► Afegeix la possibilitat que l'usuari pugui interactuar amb el ratolí per a modificar els angles que giren la càmera respecte els eixos Y i X (en les transformacions d'Euler). Quan l'usuari mou el ratolí en horitzontal de dreta a esquerra la càmera s'ha de moure sobre l'esfera de visió en direcció a la dreta. Quan l'usuari mou el ratolí en vertical de baix a dalt la càmera s'ha de moure sobre l'esfera de visió en direcció cap a dalt.

5. ► Afegeix a la implementació del teu programa la possibilitat de fer zoom-in (amb la tecla Z) i zoom-out (amb la tecla X) de manera que es modifiqui l'angle FOV de la càmera perspectiva.

6. ► Com a exercici final d'aquest bloc, modifica el teu codi (pots fer una còpia i guardar el que tenies) de manera que la teva aplicació pinti una escena que té un terra centrat a l'origen de mida 4x4, amb 2 patricios que estan escalats de manera que la seva alçada és 1 i posicionats sobre el terra de manera que un té el centre de la base de la seva capsa contenidora al punt (1,0,1) i l'altre el té al punt (-1,0,-1). El primer Patricio estarà mirant cap a les Z positives (sense cap rotació) i l'altre estarà mirant cap a les Z negatives. Cal tenir definida una càmera perspectiva que vegi l'escena sencera, centrada, sense deformació i ocupant el màxim del viewport (fixa't que les mides de l'escena sencera són conegudes, per tant són coneguts els punts de la seva capsa contenidora).