Examen parcial d'IDI 2018-2019, Q2. (Perm: A) 8/4/2019 14:00

Nom i cognoms: Temps total: 1h 50'

**Normativa preguntes curtes**

1. Responeu les següents preguntes en el mateix full de l'enunciat.
2. Cal que les respostes siguin **clares, precises i concises**.
3. No es poden usar apunts ni calculadores ni cap dispositiu electrònic.

**Escena 1**: Una escena està formada per un terra quadrat de 20x20 ubicat en el pla XZ, centrat a l'origen de coordenades i amb costats paral·lels als eixos de coordenades, 3 Patricios d'alçada 4 ubicats amb el centre de la base de la seva capsa mínima contenidora en els punts (4,0,-6), (-4,0,-6) (0,0,-6) respectivament, mirant en direció X+ i un legoman d'alçada 4 amb el centre de la base de la seva capsa en (0,0,0) i mirant en direcció Z-. Tant el legoman com els Patricios estan escalats uniformement.

1. *(1 punt)* Tenint en compte l'escena 1, indica el pseudo-codi necessari per a realitzar el càlcul de la matriu VM mitjançant angles d'Euler per a què la imatge que s'obtingui sigui la mateixa que amb una VM construïda fent: `VM = lookAt ((15,2,-6), (0,2,-6), (0,1,0))`. *Nota: L'òptica (PM) és perspectiva i no la modifiquem.*

   **Solució:**

```
VM = Translate (0,0,-15);
VM = VM* G_y (-90);
VM = VM * Translate (0,-2,6);
```

2. *(1 punt)* Tenint en compte que els punts mínim i màxim de la capsa contenidora del model del Patricio són: (Pminx, Pminy, Pminz) i (Pmaxx, Pmaxy, Pmaxz), i que el model mira inicialment en direcció Z+, completa la funció següent que calcula i retorna la TG necessària per a pintar el primer Patricio de l'escena 1, que és el que té la seva base centrada al punt (4, 0, -6). Recorda que ha de tenir alçada 4 i mirar en direcció X+:

```
tg_Pat1 () {
        ...
        ...
        ...
    return (TG);
}
```

**Solució:**

```
tg_Pat1 () {
   escala = 4/(Pmaxy-Pminy)
   cbase = ((Pminx+Pmaxx)/2, Pminy, (Pminz+Pmaxz)/2))
   TG = translate (4, 0, -6)
   TG = TG * rotate_Y (90)
   TG = TG * scale (escala, escala, escala)
   TG = TG * translate (-cbase)
   return (TG);
}
```

3. *(1 punt)* Tenim una escena formada per un cup de costat 1 amb el vèrtex de coordenades mínimes a l'origen de coordenades i totes les cares del mateix color. Completa els paràmetres d'una càmera ortogonal que permeti veure centrat al viewport un hexàgon regular (polígon de 6 costats iguals). L'única restricció al window és que no deformi i permeti veure tot el polígon. El viewport és quadrat.

```
VM = lookAt ( (2,2,2) , (0.5,0.5,0.5), (0,1,0));
PM = ortho ( -1 , 1 , -1 , 1 , sqrt(3) , 2*sqrt(3) );
```

4. *(1 punt)* Tenim un dibuix de color groc amb saturació màxima (S=1) i intensitat 0.6 (B=0.6).

   a) Quina codificació té aquest color en el format RGB?
   **Solució:** RGB = (0.6, 0.6, 0)

   b) Si mirem aquest dibuix a través d'un filtre de color cian (que només deixa passar la llum de color cian), de quin color veurem el dibuix?
   **Solució:** El veurem de color verd (RGB = (0, 0.6, 0)).

5. *(0.5 punts)* Quin és el nombre de VAOs i VBOs que hem de crear per a poder pintar l'escena 1 si per al model de cada objecte volem guardar les coordenades i el color dels vèrtexs?

```
VAOs = 3 ;
VBOs = 6 ;
```

6. *(0.5 punts)* Completa el codi del Vertex Shader següent per a què `gl_Position` tingui les coordenades que s'esperen a la sortida del Vertex Shader:

```
in vec3 vertex;
uniform mat4 VM, PM, TG;

void main()
{
   gl_Position = PM * VM * TG * vec4(vertex, 1) ;
}
```

7. *(0.5 punts)* Ordena els següents processos del procés de visualització d'OpenGL:

```
i.   Obtenció de coordenades d'observador          1) ...
ii.  Rasterització                                 2) ...
iii. Obtenció de coordenades de dispositiu         3) ...
iv.  Obtenció de coordenades normalitzades         4) ...
```

**Solució:** `1) i.  2) iv.  3) iii.  4) ii.`

8. *(0.5 punts)* Dibuixem un quadrat mitjançant dos triangles amb vèrtexs V1=(-1,-1,0), V2=(1,-1,0), V3=(-1,1,0), V4=(-1,1,0), V5=(1,-1,0) i V6=(1,1,0). Suposant que el viewport és quadrat, dibuixa què es veurà en el viewport si pintem aquest quadrat amb els següents vertex i fragment shaders (els colors els pots deixar indicats).

Vertex Shader:

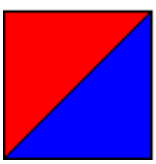
```
in vec3 vertex;
uniform mat4 VM, PM, TG;

void main()
{
   gl_Position = vec4(vertex,1);
}
```

Fragment Shader:

```
void main ()
{
   if (gl_FragCoord.x < gl_FragCoord.y)
      gl_FragColor = vec4 (1, 0, 0, 1);
   else
     gl_FragColor = vec4 (0, 0, 1, 1);
}
```

**Solució:**



Examen parcial d'IDI 2018-2019, Q2. (Perm: A) 8/4/2019 14:00

Nom i cognoms:

**Normativa del test**

(a) A les graelles que hi ha a continuació, marca amb una creu les teves respostes de l'examen. **No es tindrà en compte cap resposta fora d'aquestes graelles**.

(b) No es poden usar apunts, calculadores ni cap dispositiu electrònic.

(c) Totes les preguntes tenen una única resposta correcta.

(d) Les preguntes contestades de forma errònia tenen una **penalització del 33%** del valor de la pregunta.

| Num | A | B | C | D |
|---|---|---|---|---|
| 9 | | **X** | | |
| 10 | **X** | | | |
| 11 | | | | **X** |
| 12 | **X** | | | |

| Num | A | B | C | D |
|---|---|---|---|---|
| 13 | **X** | | | |
| 14 | | **X** | | |
| 15 | **X** | | | |
| 16 | | **X** | | |

9. *(0.5 punts)* Tenint en compte la descripció de l'escena 1 feta en el full 1 de l'examen i considerant com a posicionament de càmera: OBS=(15,2,-6); VRP=(0,2,-6); i up=(0,1,0) (ja vista en un exercici anterior). Què es veuria en el viewport si tenim una òptica ortogonal amb znear=1, zfar=30 i window=(-3,3,-3,3)? El viewport és quadrat.

a) Veuríem els tres Patricios en profunditat i el terra.

b) Veuríem un únic Patricio i no veuríem el terra.

c) Veuríem el legoman davant dels tres Patricios i el terra.

d) Veuríem un únic Patricio i un legoman, no veuríem el terra.

10. *(0.5 punts)* El procés que obté les coordenades de dispositiu d'un vèrtex dins del procés de visualització:

a) Només requereix el viewport i les coordenades normalitzades del vèrtex.

b) Requereix el vèrtex en coordenades de clipping, el window de la càmera i el viewport.

c) Les dades que requereix depenen de si es realitza abans o després del Fragment Shader.

d) Podem triar si es realitza abans o després del clipping.

11. *(0.5 punts)* Quan definim la posició i orientació de la càmera (viewMatrix) mitjançant angles d'Euler:

a) No es pot emular l'efecte del vector up que tenim quan la definim amb lookAt.

b) Sempre és equivalent a tenir un up=(0,1,0).

c) Cal utilitzar lookAt per ubicar la càmera i lookAt seguit de transformacions d'Euler quan tenim interacció.

d) Modificant l'angle d'Euler de gir respecte Z podem obtenir l'efecte del vector up.

12. *(0.5 punts)* Sabem que a una impressora CMY li falla la tinta magenta (però continua imprimint). Si en imprimir un dibuix en aquesta impressora, en el dibuix imprès es veuen dues franges (F1 i F2) de colors F1 = cian i F2 = groc, de quins colors són les franges del dibuix original?

    a) F1 = blau i F2 = groc
    b) F1 = vermell i F2 = groc
    c) F1 = blau i F2 = blau
    d) F1 = vermell i F2 = cian

13. *(0.5 punts)* Es disposa d'una funció pintacub() que envia a pintar el VAO d'un cub d'aresta 10 amb el centre del cub a l'origen de coordenades. Tenim una càmera definida amb OBS= (0,0,20), VRP=(0,0,0), up=(0,1,0) Quin dels següents codis és correcte per visualitzar una escena formant una L amb el cub a la cantonada de la L centrat a l'origen de coordenades. Nota: el vertex shader està correctament definit i rep com uniform una matriu TG de 4x4; l'òptica és ortogonal i està definida també correctament.

    a)
    ```
    TG=I; TG= TG*Translació (0,0,10); modelMatrix (TG)
    Pintacub()
    TG=I; TG= TG*Translació (0,10,0); modelMatrix (TG)
    Pintacub()
    TG=I; TG= TG*Translació (10,0,0); modelMatrix (TG)
    Pintacub()
    ```
    b)
    ```
    TG1=I; TG1= TG1*Translació (0,0,10); modelMatrix (TG1)
    TG2=I; TG2= TG2*Translació (0,10,0); modelMatrix (TG2)
    TG3=I; TG3= TG3*Translació (10,0,0); modelMatrix (TG3)
    Pintacub()
    Pintacub()
    Pintacub()
    ```
    c)
    ```
    Pintacub()
    TG=I; TG= TG*Translació (0,0,10); modelMatrix (TG)
    Pintacub()
    TG=I; TG= TG*Translació (0,10,0); modelMatrix (TG)
    Pintacub()
    TG=I; TG= TG*Translació (10,0,0); modelMatrix (TG)
    ```
    d)
    ```
    TG=I; TG= TG*Translació (0,0,10); modelMatrix (TG)
    Pintacub()
    TG= TG*Translació (0,10,0); modelMatrix (TG)
    Pintacub()
    TG= TG*Translació (10,0,0); modelMatrix (TG)
    Pintacub()
    ```

14. *(0.5 punts)* Aquesta indicació de progrés...


    a) No indica res, no dóna cap informació.
    b) No és útil si la feina a fer triga més de 15 segons.
    c) Serveix sempre perquè indica que està treballant.
    d) Cap de les anteriors.

15. *(0.5 punts)* Tenint en compte que defineix una aplicació per indicar l'hora, quin tipus de representació es correspon a la icona de la imatge?


a) Similaritat

b) Simbòlic

c) Exemple

d) Arbitrari

16. *(0.5 punts)* Del següent llistat d'afirmacions relatius a l'us de colors en una interfície gràfica, indica quina és FALSA:

a) El colors saturats poden provocar fatiga visual.

b) Es millor seleccionar molts colors diferents per a les icones d'una interfície, per a que sembli més senzill distingir els elements.

c) Per a que es vegi bé un text, es recomanable utilitzar contrast, per exemple obscur per al text, i suau per al fons.

d) Si tenim botons del mateix color, sobre un fons que canvia d'intensitat, l'usuari els veurà diferents.