

## Interacció i Disseny d'Interfícies

Data: 26 de març de 2021

## CONTROL PARCIAL

Temps: 2h

L'examen es fa en paper, cada problema en un full separat. Feu-ne les fotos i munteu-les en un únic pdf. Useu Microsoft Lens o qualsevol altre App per fer-ho.

Format de lliurament : un ZIP anomenat **Cognom1Cognom2\_Nom.zip** que contingui:

Enunciat.pdf

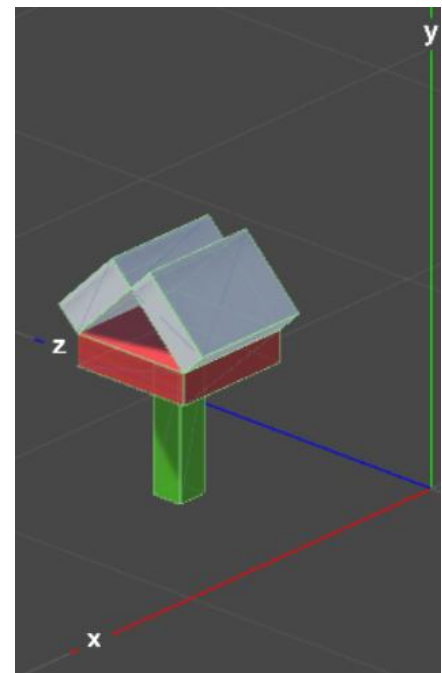
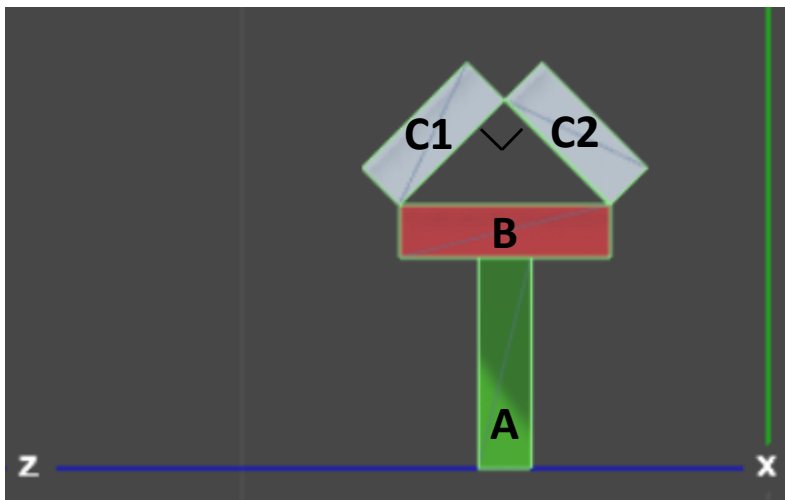
Solucio.pdf

**(1) [3,5p]** A partir de la funció **pintaCub**, que dibuixa un cub de **costat 2** centrat a l'origen, volem dibuixar l'escena mostrada a continuació:

A.- Peu verd: Prisma de base quadrada de costat 1 i alçada 4. Centre de la base a (5,0,5)

B.- Base vermella: Prisma de base quadrada de costat 4 i alçada 1. Està centrat sobre A.

C1 i C2.- "Teulades" blanques: 2 prismes de la mateixa mida, amb la dimensió petita de mida 1. Cadascun comparteix una aresta amb B. C1 i C2 comparteixen l'aresta superior i formen un angle de 90 graus entre si.



Escriu les transformacions geomètriques necessàries per passar del cub original als quatre prismes usant les funcions de la llibreria glm. Fes-ho programant les funcions **modelTransformCubX()**, on X és l'etiqueta del fragment ( A, B, C1 o C2 ):

```
void modelTransformCubA() {
    glm::mat4 TG(1.0f);
    TG = glm::translate(TG, glm::vec3(5.0, 2, 5.0));
    TG = glm::scale(TG, glm::vec3(1.0, 4, 1.0));
    TG = glm::scale(TG, glm::vec3(0.5, 0.5, 0.5));
    glUniformMatrix4fv (transLoc, 1, GL_FALSE, &TG[0][0]);
}
```

```
void modelTransformCubB() {
    glm::mat4 TG(1.0f);
    TG = glm::translate(TG, glm::vec3(5.0, 4.5, 5.0));
    TG = glm::scale(TG, glm::vec3(4.0, 1.0, 4.0));
    TG = glm::scale(TG, glm::vec3(0.5,0.5,0.5));
    glUniformMatrix4fv (transLoc, 1, GL_FALSE, &TG[0][0]);
}
```

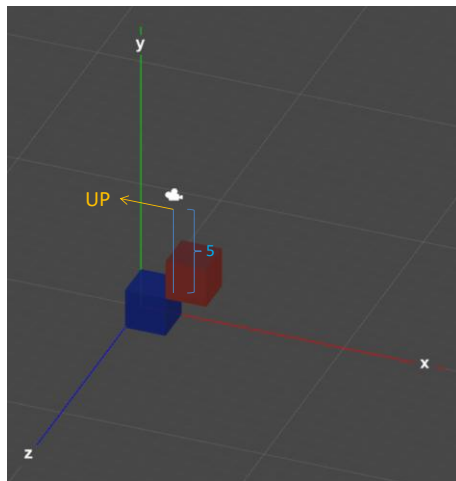
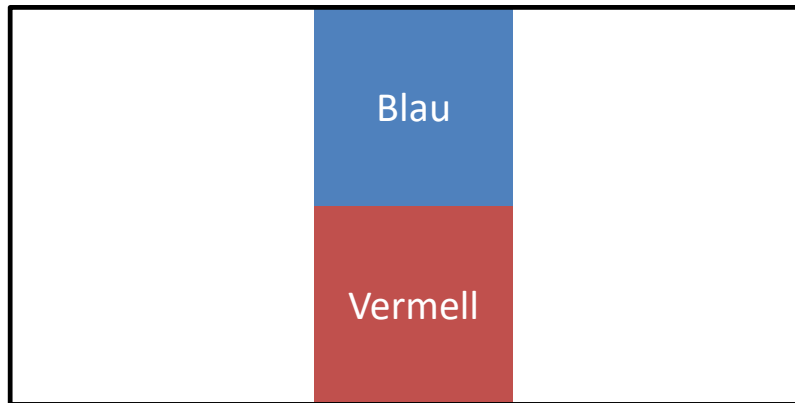
```
void modelTransformCubC1() {
    float a = 2*sqrt(2);
    glm::mat4 TG(1.0f);
    TG = glm::translate(TG, glm::vec3(5, 5, 5+2 ));
    TG = glm::rotate(TG, glm::radians(+45), glm::vec3(1.0, 0.0, 0.0));
    TG = glm::translate(TG, glm::vec3(0.0, 0.5, -a/2));
    TG = glm::scale(TG, glm::vec3(4.0, 1.0, a));
    TG = glm::scale(TG, glm::vec3(0.5,0.5,0.5));
    glUniformMatrix4fv (transLoc, 1, GL_FALSE, &TG[0][0]);
}
```

```
void modelTransformCubC2() {
    float a = 2*sqrt(2);
    glm::mat4 TG(1.0f);
    TG = glm::translate(TG, glm::vec3(5, 5, 5-2 ));
    TG = glm::rotate(TG, glm::radians(-45), glm::vec3(1.0, 0.0, 0.0));
    TG = glm::translate(TG, glm::vec3(0.0, 0.5, a/2));
    TG = glm::scale(TG, glm::vec3(4.0, 1.0, a));
    TG = glm::scale(TG, glm::vec3(0.5,0.5,0.5));
    glUniformMatrix4fv (transLoc, 1, GL_FALSE, &TG[0][0]);
}
```

**(2)** Una escena està formada per dos cubs de costat 2, un de color blau i un de color vermell. El cub **blau** reposa sobre els eixos de coordenades, amb la seva coordenada mínima en el (0,0,0). El cub **vermell** té les seves cares paral·leles als plans coordenats, i el seu centre està a la coordenada (3,3,1).

**(2.1)** [2,5p] Calcula **tots els paràmetres** d'una càmera amb projecció ortogonal que visualitzi l'escena centrada segons mostra la figura més a sota (és un

viewport de 800x600). Restriccions: la càmera ha d'estar enfocada cap a les y negatives i a una distància de 5 del centre de l'escena.



#### Paràmetres de visió:

Centre\_escena = (2,2,1)

OBS = Centre\_escena+(0,5,0) = (2,7,1)

VRP = Centre\_Escena = (2,2,1)

UP=(-1,0,0)

#### Finestra de projecció:

zN=5-2=3

xF=5+2=7

RA\_w= w/h = 2/4 =0.5

RA\_v=800/600=1.3333

RA\_w<RA\_v → cal augmentar w i mantenir h

Així doncs, h=4, i recalculem w :

w\*=h\*RA\_v = 4\*1.3333=5.3333

Com que la finestra de visió és simètrica

$$R = -L = 5.333/2 = 2.666666$$

$$T = -B/2 = 2$$

```
glm::mat4 PM = glm::ortho(L,R,B,T,zN,zF)
```

```
glm::mat4 PM = glm::ortho(-2.66,2.66,-2,2, 3, 7)
```

**(2.2) [1,5p]** Calcula la ViewMatrix que correspondria a aquesta càmera manualment a partir de transformacions geomètriques.

```
glm::mat4 VM(1.0f);
```

```
VM = glm::translate(VM , glm::vec3(0,0,-5)); // traslladem -distància a càmera en z
```

```
VM = glm::rotate(VM , glm::radians(+90), glm::vec3(1,0,0)); // rotació en x, ->theta=90
```

```
VM = glm::rotate(VM , glm::radians(-90), glm::vec3(0,1,0)); // rotació en y, ->psi=90
```

```
VM = glm::translate(VM , glm::vec3(-2,-2,-1)); // traslladem -VRP
```

**(2.3) [1p]** Dona els paràmetres de projecció en perspectiva (FOV, ra, zN i zF) per visualitzar l'escena des de qualsevol angle a una distància de 5 al centre de l'escena en un viewport de 1200x400.

$$P_{min} = (0,0,0)$$

$$P_{max} = (4,4,2)$$

$$R = \sqrt{(4^2 + 4^2 + 2^2)} / 2 = 3$$

$$d = 5$$

$$zN = d - R = 2$$

$$zF = d + R = 7$$

$$\alpha_v = \arcsin(R/d) = \arcsin(3/5) = 36.86^\circ$$

$$FOV = 2 \alpha_v = 73.74^\circ$$

$RA_w = 1$  (recordeu, una esfera es projecta en un quadrat en qualsevol direcció)

$$RA_v = 1200/400 = 3$$

Donat que  $RA_v > RA_w$  cal augmentar  $RA_w$ . Això implica mantenir alçada (mantenir FOV) i modificar  $RA_w$ :

$$RA_w^* = RA_v = 3$$

```
glm::mat4 PM = glm::perspective(glm::radians(73.74), 3, 2, 7);
```

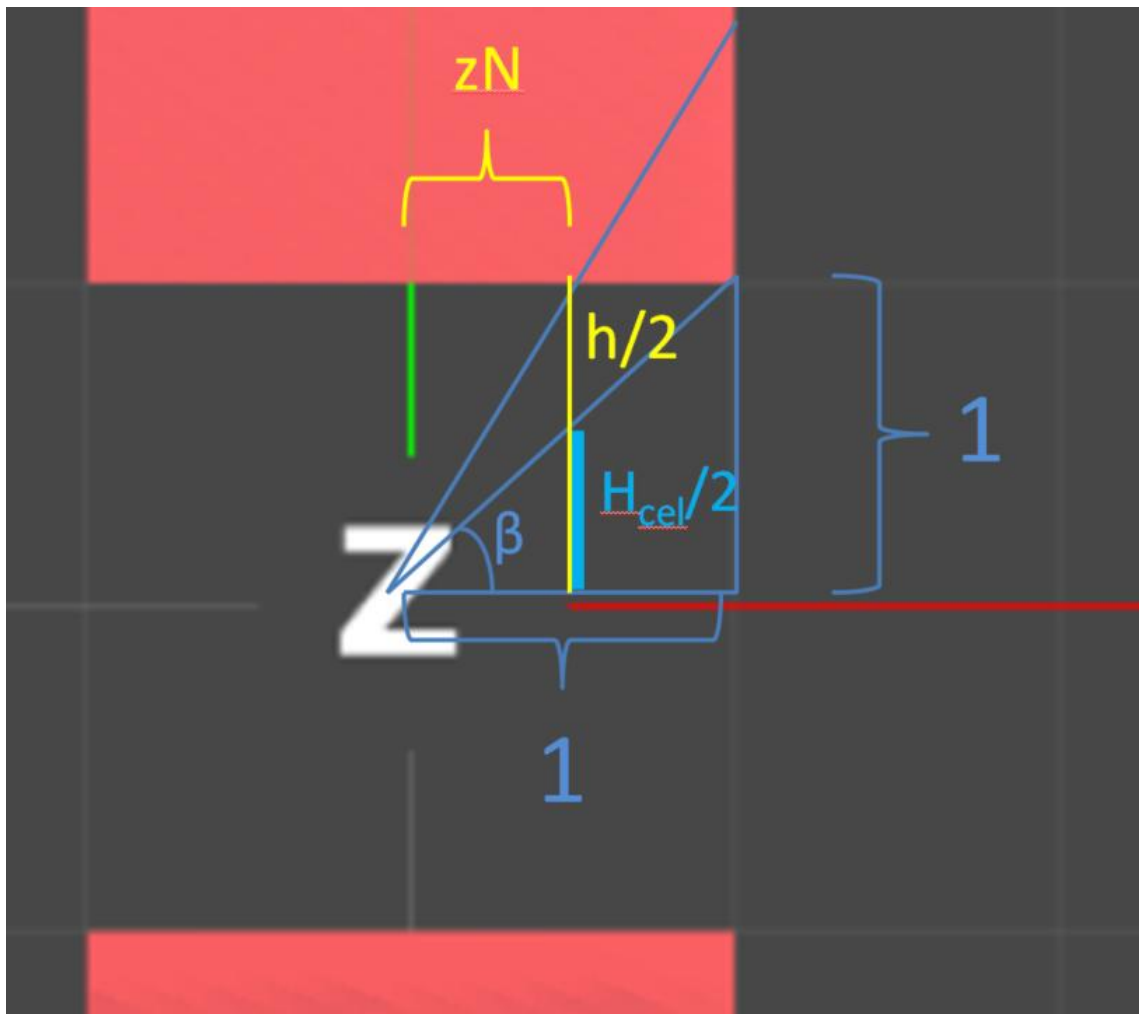
.....

**(3)** [1,5p] Donada una escena formada per dos cubs vermells de costat 2 amb les seves cares paral·leles als plans coordenats, l'un centrat a  $(0,2,0)$  i l'altre a  $(0,-2,0)$ . Tenim la càmera situada a l'origen de coordenades, el  $VRP = (10,0,0)$  i  $\overrightarrow{UP} = (0,1,0)$ .

Calculeu FOV, zNear i zFar d'una càmera de **perspectiva** per tal que es mostri la següent escena (no cal calcular la relació d'aspecte):



El cub superior i l'inferior ocupen cadascun un terç de l'alçada del viewport. L'altre terç és el "buit" central.



Angle de visió del "cel" :  $\beta = \text{atan}(1/1) = 45^\circ$

$$h_{\text{cel}}/2 = z_{\text{Near}}$$

h: alçada de la finestra de projecció

Com que  $h_{\text{cel}}$  ha de ser 1/3 de l'alçada de la finestra:  $h_{\text{cel}} = h/3$

$$h/6 = z_{\text{Near}} \rightarrow h = 6 \cdot z_{\text{Near}}$$

$\alpha_v$  determina h:

$$\tan(\alpha_v) = (h/2) / z_{\text{Near}}$$

com que  $h = 6 \cdot z_{\text{Near}}$

$$\tan(\alpha_v) = 6/2 = 3$$

$$\alpha_v = \text{atan}(3) = 71.56^\circ$$

$$\text{FOV} = 2 \cdot \alpha_v = 143.13^\circ$$

$h=2$  (la distància del cub superior a l'inferior)

$$z_{\text{Near}} = h/6 = 2/6 = 0.3333 \text{ ( o menor! )}$$



**zFar qualsevol valor >1**