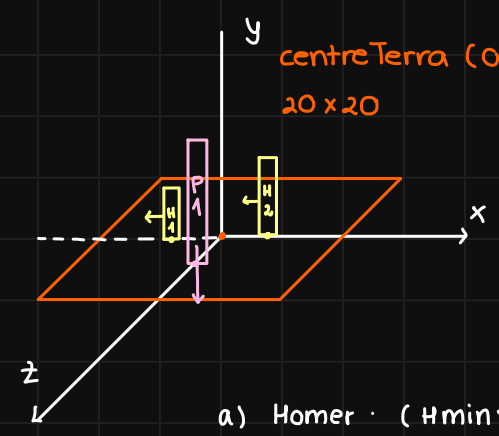
notebook

TOPIC: PARCIAL 2022 - 2023 Q1 DATE:



centreTerra (0,0,0)

20 x 20

H1 → altura 4 ; centre (-8,0,0), -x

H2 → altura 8 ; centre (8,0,0); -x

P1 → altura 10; centre (0,0, 7.5); +z

a) Homer · (Hminx, Hminy, Hminz), (Hmaxx, Hmaxy, Hmaxz), +z ⇒ -x

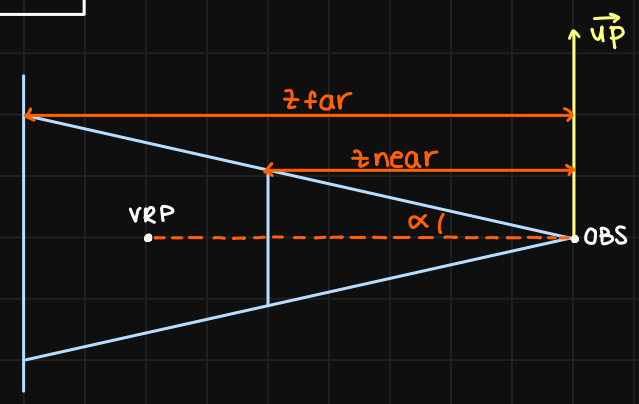
```
glm::mat4 TGHomer (centre, alcada) {
    TG = I;   // resetegem TG
    escala = alcada / (Hmaxy - Hminy),
    cBase = ((Hminx + Hmaxx)/2, Hminy, (Hminz + Hmaxz)/2);
    TG = TG * translate (centre);
    TG = TG * rotate (-90, (0,1,0));
    TG = TG * scale (escala, escala, escala);
    TG = TG * translate (-cBase);
    return (TG);
}
```

b)

```
TG1 = TGHomer ((-8,0,0), 4)
modelMatrix (TG1);
pintaModel (Homer);
TG2 = TGHomer ((8,0,0), 8)
modelMatrix (TG2);
pintaModel (Homer);
TG3 = TGPatricio ((0,0,7.5), 10)
modelMatrix (TG3);
pintaModel (Patricio);
TG4 = I;
modelMatrix (TG4);
pintaModel (Terra);
```

TOPIC: DATE:

CAMARA



- VRP = centro de la escena (caja contenedora)
- OBS = punto de origen de la cam ⇒ OBS = VRP + d · (0,0,1) (dir. de visión)
  (plano al que queremos que sea paralelo)
- UP = vector vertical que orienta la cam.

CAMARA PERSPECTIVA

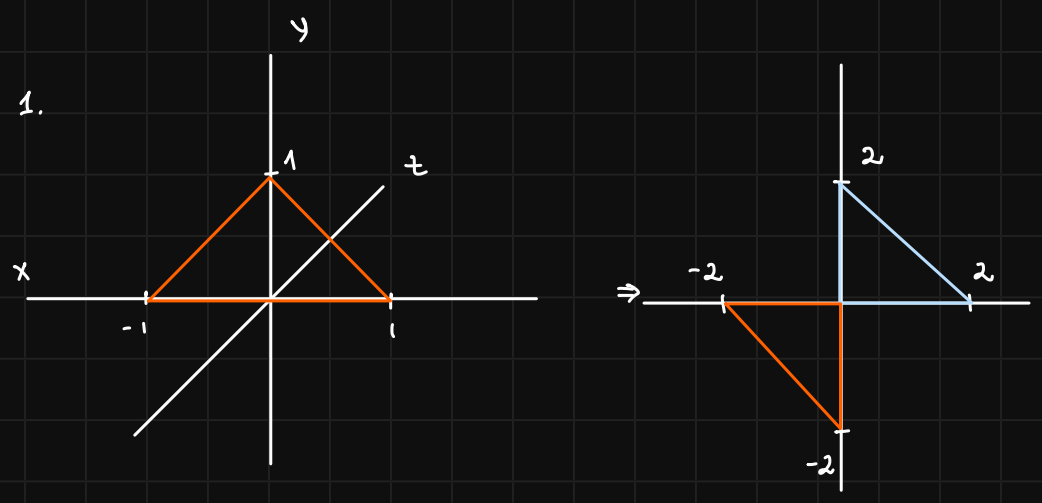
- ra_w = relación de aspecto ⇒ $ra_w = a_w / h_w$ ($h_w = 2 \cdot z_{near} \cdot tg(\alpha)$)
- Znear = distancia entre el OBS y y el inicio de la escena ⇒ $z_{near} = d - Radi$ ($d = 2R$)
- Z far = distancia entre el OBS y y el final de la escena ⇒ $z_{far} = d + Radi$
- FOV = grado de apertura de la cámara ⇒ $FOV = 2\alpha = 2 \cdot \arcsin(ra_w)$

FORMULITAS

- $\alpha = \arcsin(\frac{R}{d}) \Rightarrow FOV = 2 \arcsin(\frac{R}{d})$
- $P_{medio} = (P_{max} + P_{min}) / 2$
- $R = \sqrt{(P_{med_x})^2 + (P_{med_y})^2 + (P_{med_z})^2}$

TOPIC: DATE:

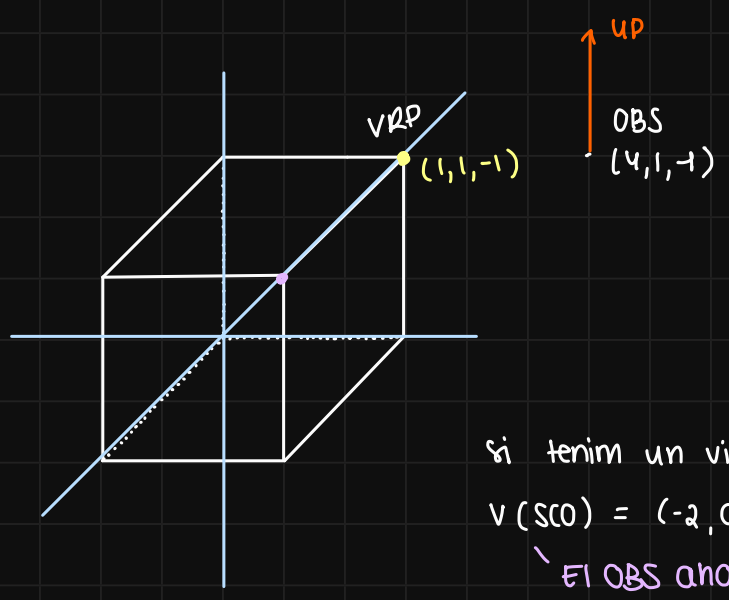
1.



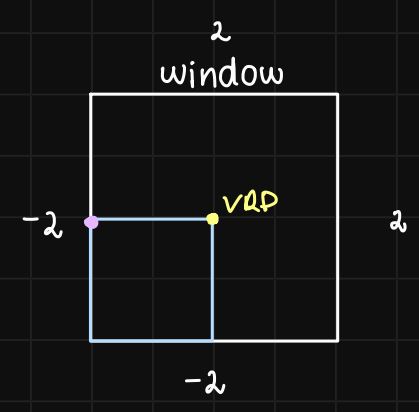

```
tg_triangle() {
    TG2 = I;
    TG2 = TG * rotate (-45, (0,0,1))
    TG2 = TG * scale (2/√2, 2/√2, 2/√2)
    TG2 = TG * translate (0,-1,0)
    return(TG2)
}
```

* mi lado vale $\sqrt{2}$ yo quiero que valga 2 ⇒ escala = $2/\sqrt{2}$

2.



⇒



OBS - VRP

d = (4,1,-1) - (1,1,-1) = (3,0,0)

znear = 3 ← costat cub

zfar = 3+2 = 5

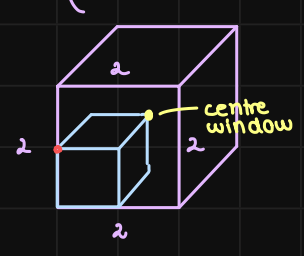
* punt mes proper i punt més llunya

Si tenim un viewport de 800 x 600 dona les coord. del cub del V(1,1,1) en

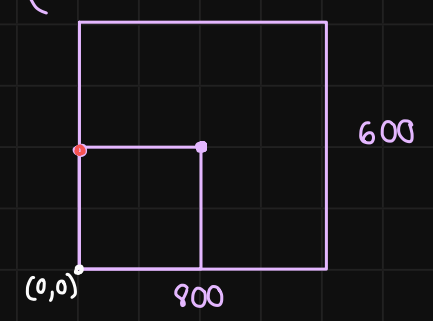
V(SCO) = (-2,0,-3)

↳ El OBS ahora es el (0,0,0)

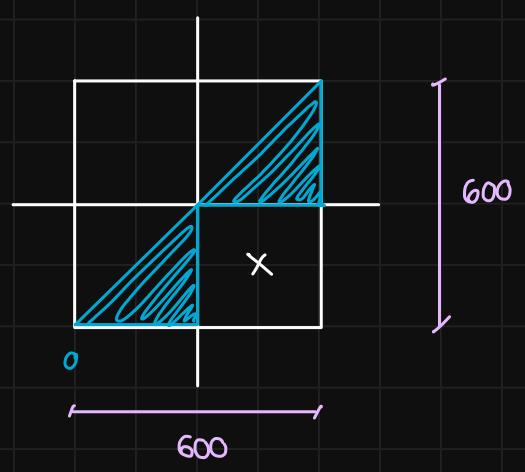
V(SCN) = (-1,0,1)



V(SCD) = (0,300)



3. Cuadrat V1(-1,-1,0), V2(1,-1,0), V3(1,1,0), V4(-1,1,0)



x > 300 + y < 300 0 y > x ⇒ color fons

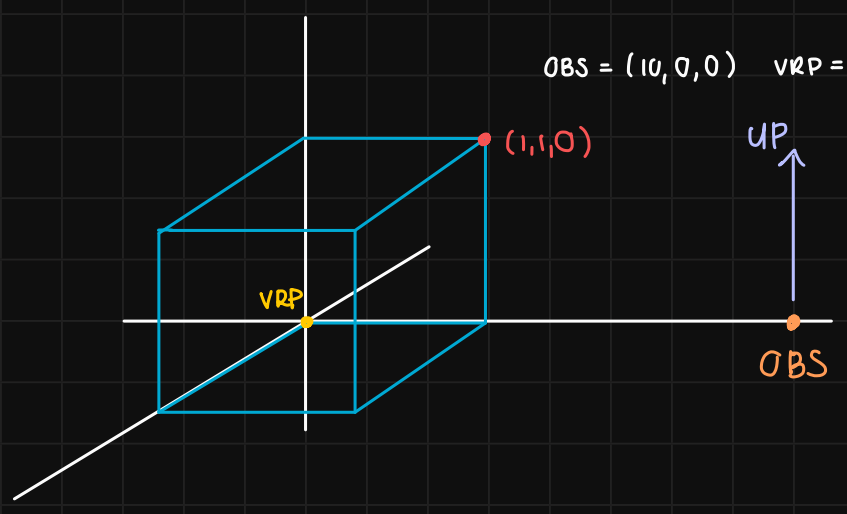
sino pinta RGB (0,0,1)

↳ Blue

4. RGB (0.7, 0.7, 0)

a) RGB (0.7, 0, 0)

5.



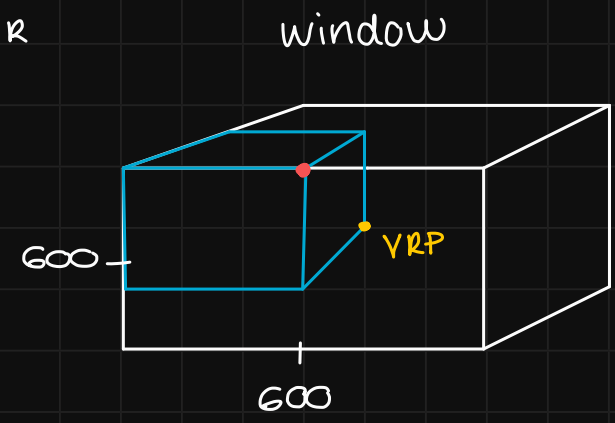
OBS = (10,0,0) VRP = (0,0,0) UP = (0,1,0)

znear = 5 → 5 = 10 - R

zfar = 15

d = 10 R = 5

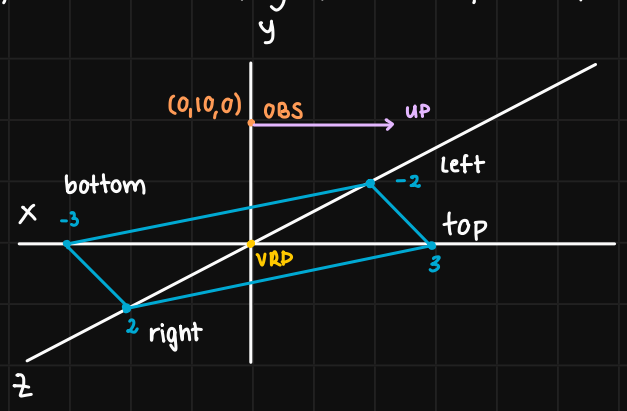
P = (300, 600)

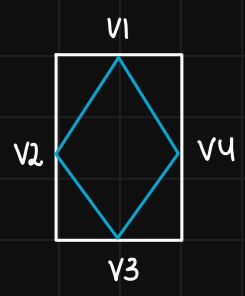


TOPIC: DATE:

6. V1(3,0,0), V2(0,0,-2), V3(-3,0,0), V4(0,0,2)

a) càmara ortogonal viewport 400 x 800





VM = lookAt (OBS (0,10,0), VRP (0,0,0), UP ...)

PM = ortho (left, right, bottom, top, 8, 12) (8 = tnear, 12 = tfar)

VM = lookAt ((0,10,0), (0,0,0), (1,0,0))

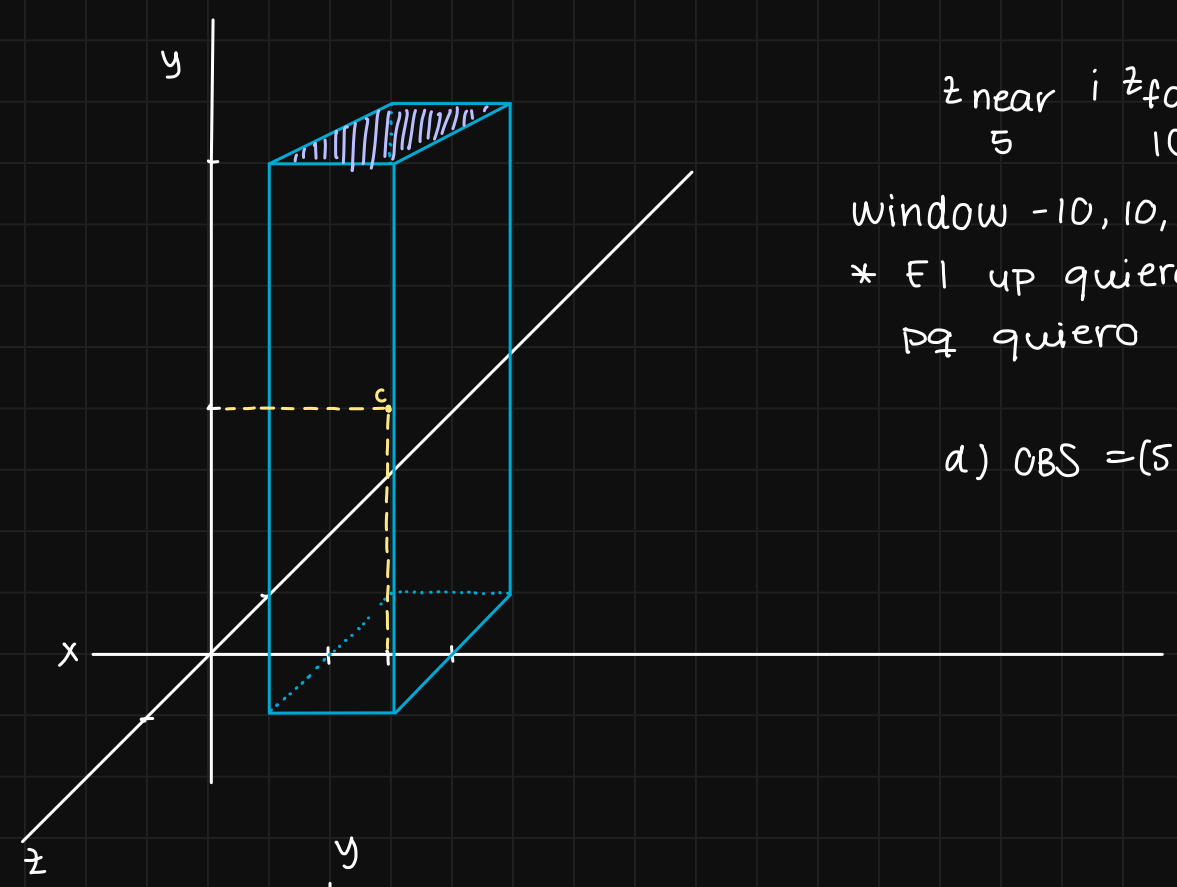
PM = ortho (-2, 2, -3, 3, 8, 12)

b) $ra_w = a_w / h_w = 2-(-2) / 3-(-3) = 2/3$

7. CMY → dibuix color verd (RGB(0,1,0)) però surt de color cian (CMY(1,0,0)), que ha passat?

CMY (1,0,1) ⇒ no tenia tinta groga.

8. centre escena (5,10,0), capsa contenidora 10 x 20 x 10 (x,y,z), usem un viewport quadrat.

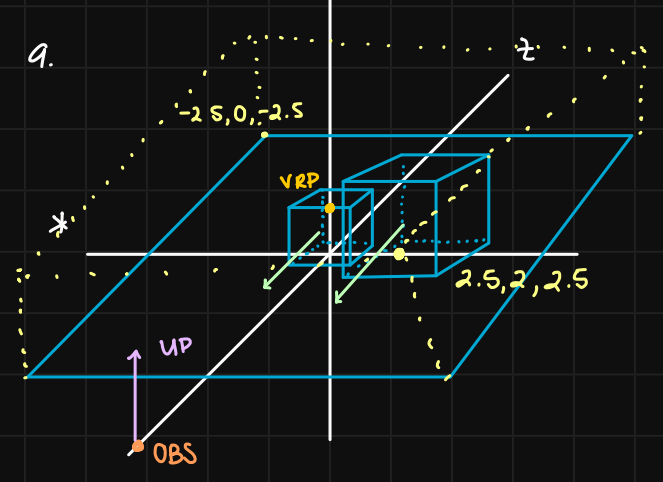


znear i zfar
5 10

window -10,10,-10,10 = cubo 5x5

* El up quiero que esté en las x o las z pq quiero ver un cuadrado

d) OBS =(5,20,0); VRP = (5,10,0); UP = (-1,0,0)



* Centrat a l'origen, 5x5
* Patricio d'alçada 1 (1x1x1), centrat al (0,0,0), Z+
* Patriùo d'alçada 2 (2x2x2), centrat al (1.5,0,0), Z+

a) Parametres càmara, esfera contenidora max. viewP.

viewPort quadrat

$P_{mitj} = \frac{(2.5,2,2.5)+(-2.5,0,-2.5)}{2} = \frac{2}{2} = (0,1,0)$

VRP (0,1,0)

UP (0,1,0)

OBS = (0,1,0) + d (0,0,1) (eix z)

FOV = 2 arcsin (R/d)

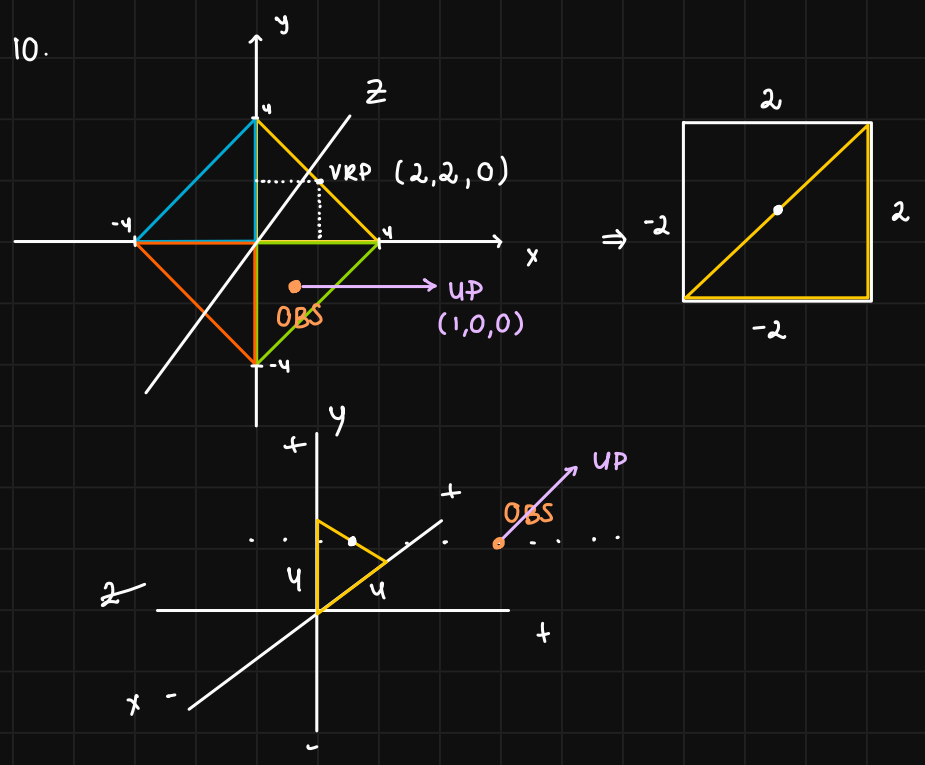
ra = 1

znear = d - R

zfar = d + R

TOPIC: DATE:

10.



VM = lookAt (OBS, VRP, UP)

PM = ortho (left, right, bottom, top, 1, 3)
-2 2 -2 2

VRP(2,2,0) → OBS = (2,2,0) + d(0,0,1)

d = 2 R

znear = d - R , 1 = d - R , d = 1 + R
zfar = d + R , 3 = d - R , d = 3 - R
} 1 + R = 3 - R

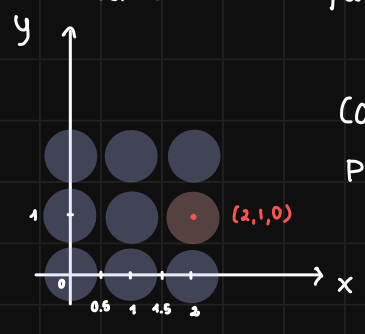
2R = 2 , R = 1

d = 2

OBS = (2,2,0) + 2 (0,0,1) = (2,2,2)

11. pintaEsfera () → pinta esfera de R = 5 centrada al 0,0,0

* usando esto queremos:



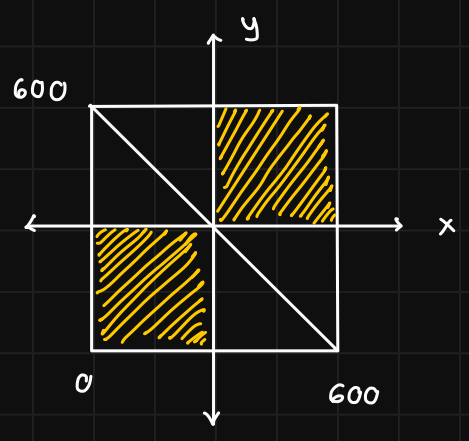
Cada esfera le R = 0.5

Plà xy

escala = 0.5 / 5 = 0.1

```
TG_vermella () {
    TG = I;
    TG = TG * translate (2,1,0);
    TG = TG * scale (0.1, 0.1, 0.1);
    return(TG);
}
```

12.




```
if ((x>300 && y>300) || (x<300 && y<300)) color = (1,1,0,1)
else color = (1,1,1,1)
```

13. nº VAO i nº VBO para hacer la Escena del Ej. 9 (2 patricios + 1 suelo)

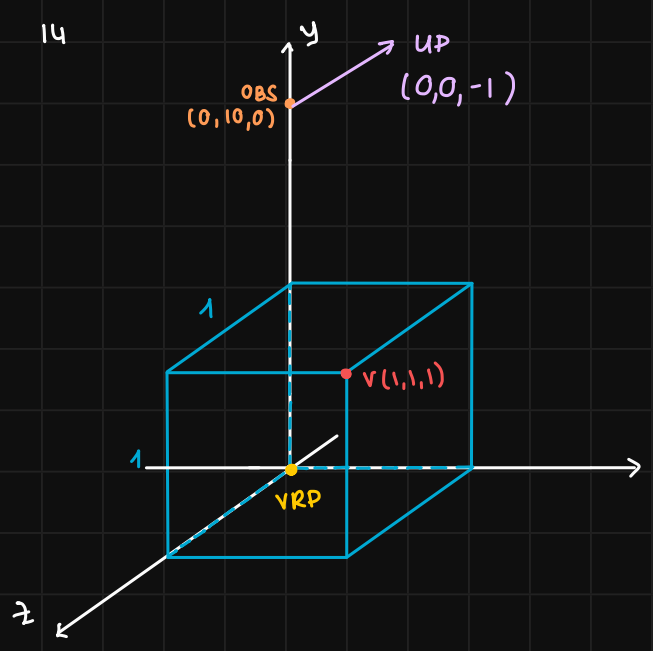
a) Per cada model volem guardar els vertexs

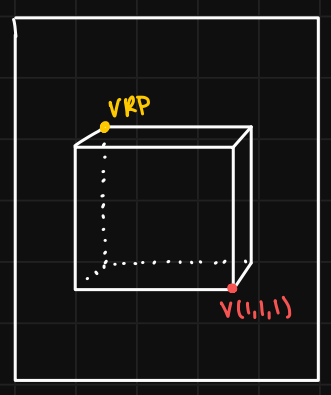
VAO = 2 VBO = 2 (1 VAO i 1 VBO x model)

b) Per cada model volem els vertexs i el color

VAO = 2 VBO = 4 (1 VAO i 2 VBO x model)

14





viewport 800 x 600

$ra_V = \frac{800}{600} = 4/3$

window