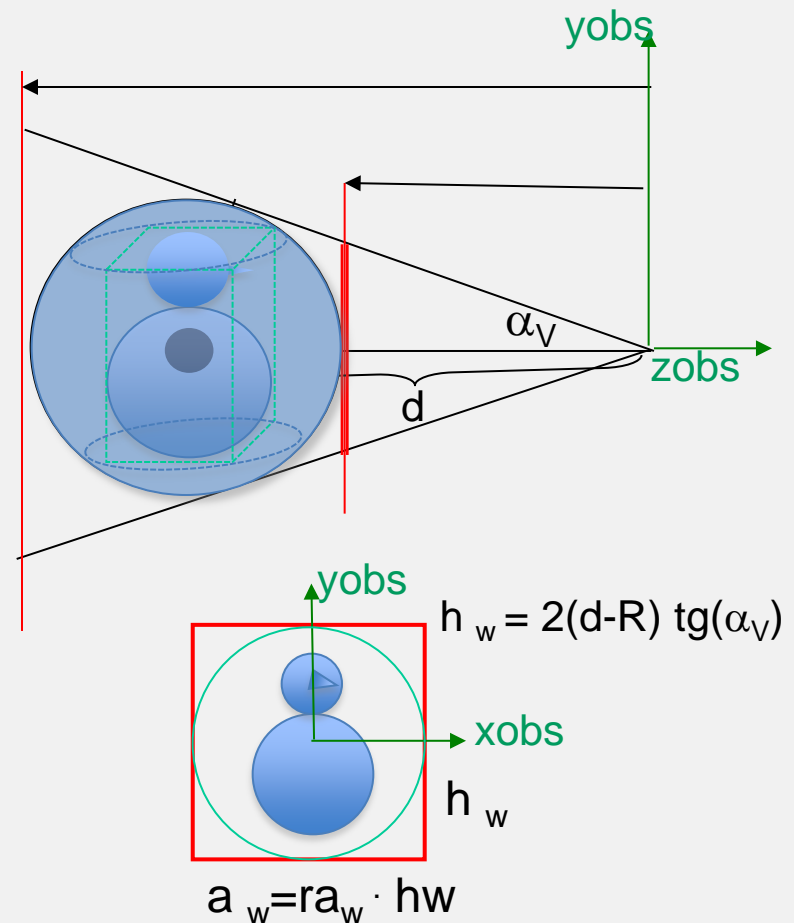
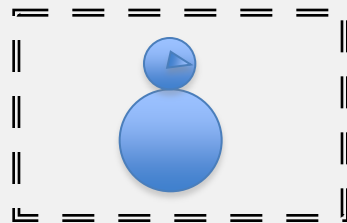
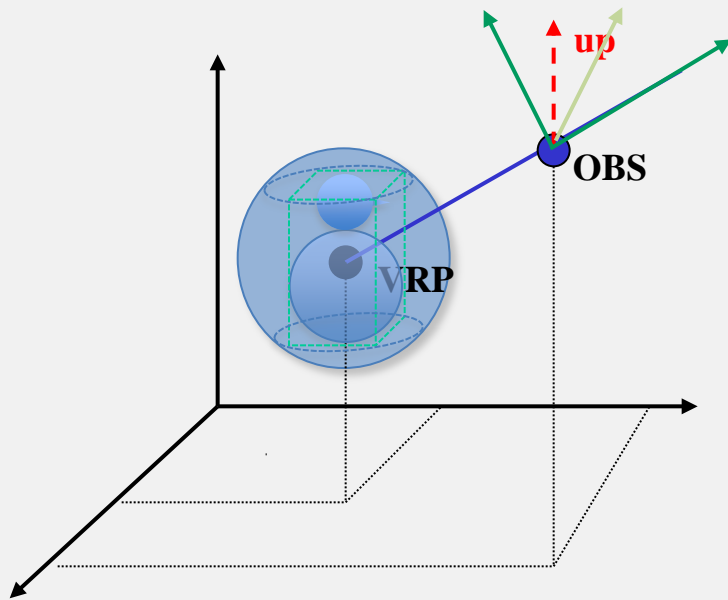


# Laboratori OpenGL – Sessió 2.2

- Càlcul càmera per a visualitzar escena (càmera 3<sup>a</sup> persona)
- Redimensionat finestra sense deformació ni retallat (resize)
- Visualitzar objecte qualsevol
- Òptica ortogonal
- *Resize* també amb òptica ortogonal

# Càmera en 3<sup>a</sup> persona (exercicis 1 i 2)

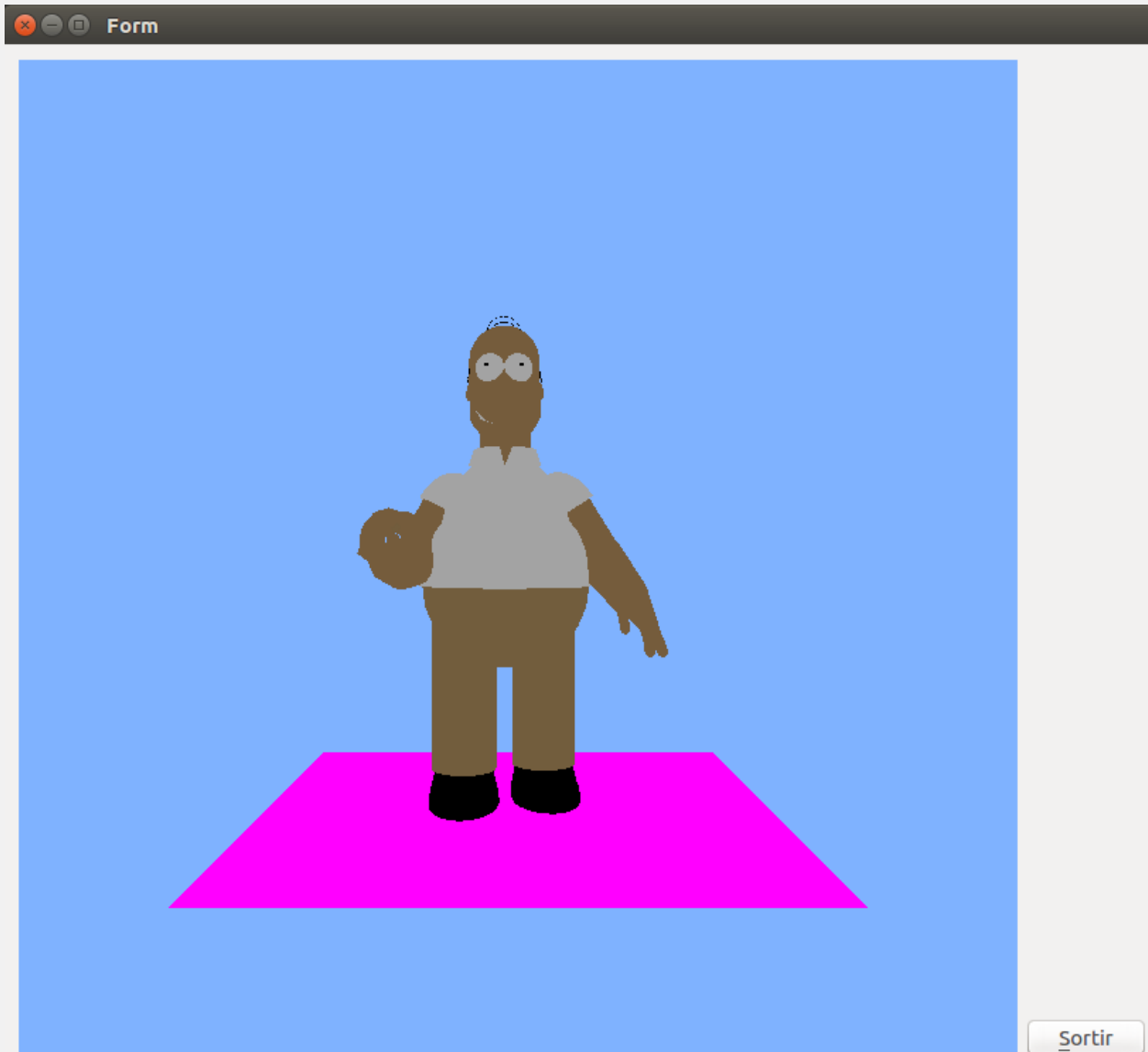
- Considerar la capsa (i esfera) mínima contenidora de l'escena
- Càlculer els paràmetres de posició i orientació (OBS,VRP,Up)
- Calcular els paràmetres de l'òptica perspectiva (FOV, raw, ZN, ZF)



# Càmera en tercera persona

- Mètode per a calcular centre i radi d'escena: (exercici 1)
  - Donats punt mínim i màxim de la caixa contenidora  
coneguts en la majoria de casos
- Usar centre i radi escena per a posar paràmetres càmera en tercera persona: (exercici 2)
  - Que es vegi escena centrada, sencera, sense retallar i ocupant màxim del viewport.

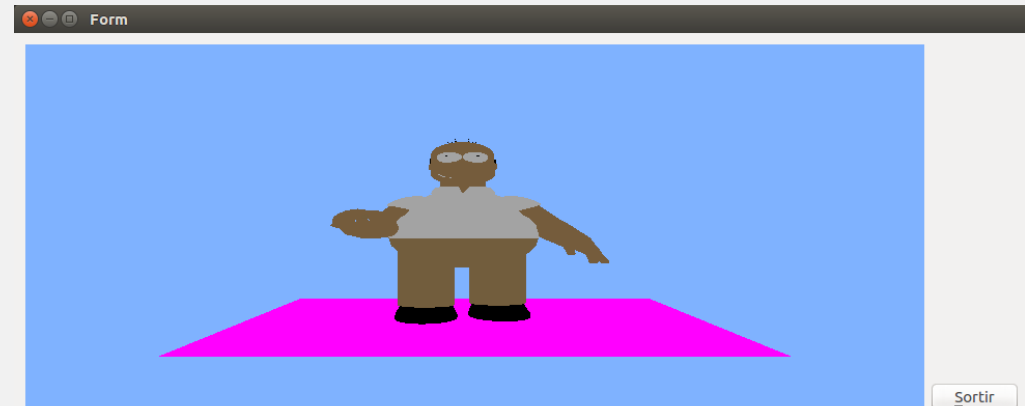
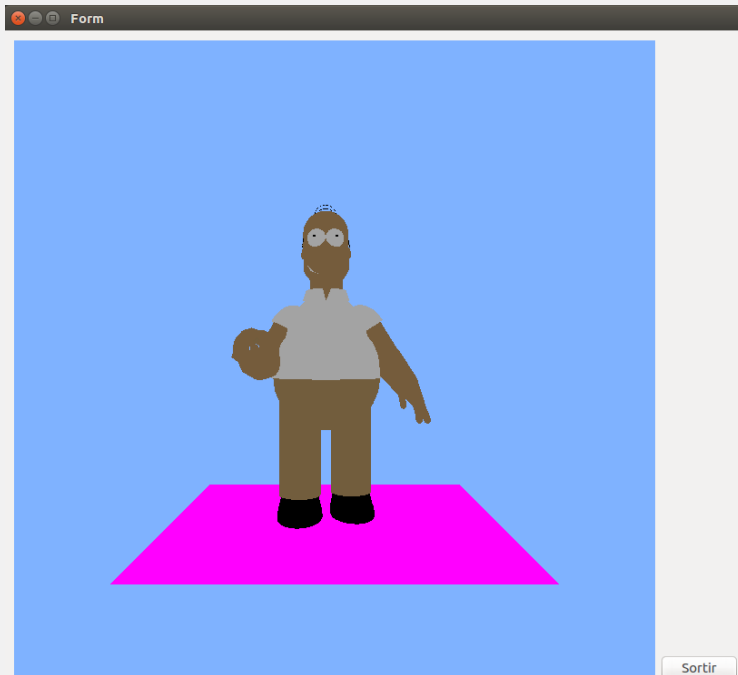
# Càmera en tercera persona



# Redimensionat sense deformació ni retallat

## (exercici 3)

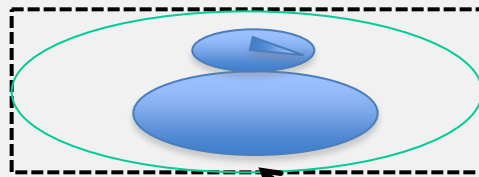
- Quan l'usuari redimensiona la finestra gràfica s'executa automàticament el mètode `resizeGL ()`
- Si aquest mètode no fa res:



# Redimensionat sense deformació ni retallat

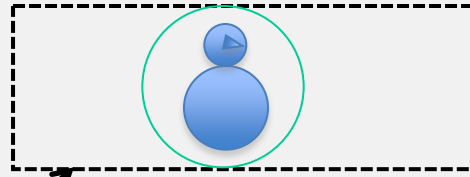
- La relació d'aspecte ( $ra$ ) del window ha de ser igual que la del viewport:  
 $ra_w = ra_v$
- Per tant si canvia la  $ra_v \rightarrow$  ha de canviar la  $ra_w \rightarrow$  refer perspective (...)

- Si  $ra_v > 1$  i  $ra_w = ra_v \Rightarrow$  la nova  $a_w^* > a_w$  mínima requerida  $\Rightarrow$  No es retalla



Amb  $ra_w=1$

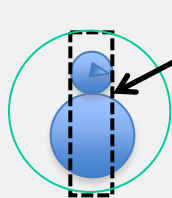
Viewport  
 $ra_v > 1$



Amb  $ra_w^*=ra_v$

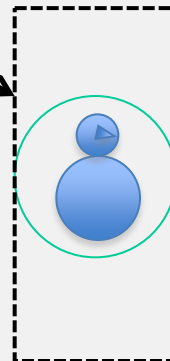
no cal modificar  $\alpha_v$  (FOV)

- Si  $ra_v < 1 \Rightarrow ra_w^* < ra_w \Rightarrow a_w^* < a_w \Rightarrow$  retallarà; per evitar-ho cal incrementar l'angle d'obertura (quedarà espai lliure a dalt i a baix)



Amb  $ra_w=ra_v$

viewport



- Amb  $ra_w = ra_v$  i nou FOV

- $FOV = 2 \alpha_v^*$  on  $\alpha_v^* = \arctg(\tg(\alpha_v) / ra_v)$

- Sempre cal calcular el nou angle a partir de l'inicial (*window* quadrat).

# Redimensionat sense deformació ni retallat

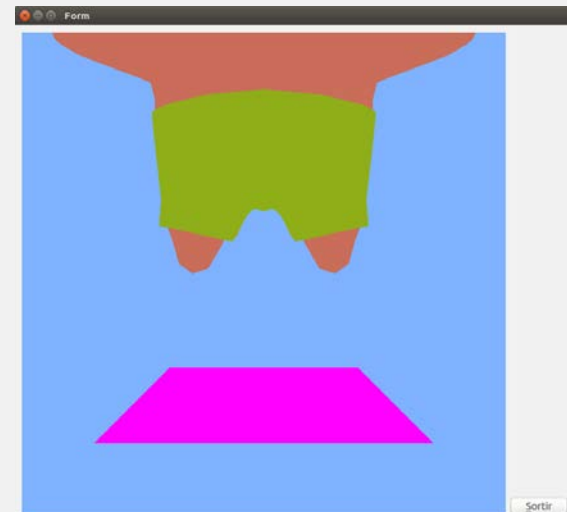
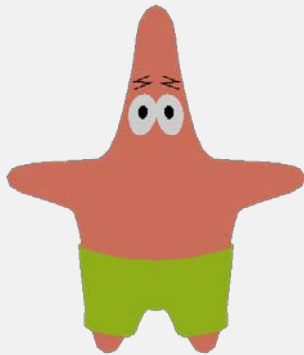
## (exercici 3)

- El mètode `resizeGL` rep com a paràmetres l'amplada i alçada de la finestra gràfica
  - `void resizeGL (int w, int h);`  
  
// possible càlcul de la relació d'aspecte del viewport  
`float ra = float (w) / float (h);`
- Mètodes de `QOpenGLWidget` que ens poden ser útils:
  - `width ()` → retorna amplada de la finestra gràfica (int)
  - `height ()` → retorna alçada de la finestra gràfica (int)

# Pintar objecte qualsevol

## (exercici 4)

- Pintem el Patricio.obj
  - Model no centrat a l'origen i de mides no controlades (decisió del dissenyador del model)
  - Cal calcular la capsula contenidora del model
  - Es vol l'objecte **escalat per a que faci alçada 4 i amb la seva base centrada a l'origen** de coordenades
  - Cal afegir transformacions de model necessàries





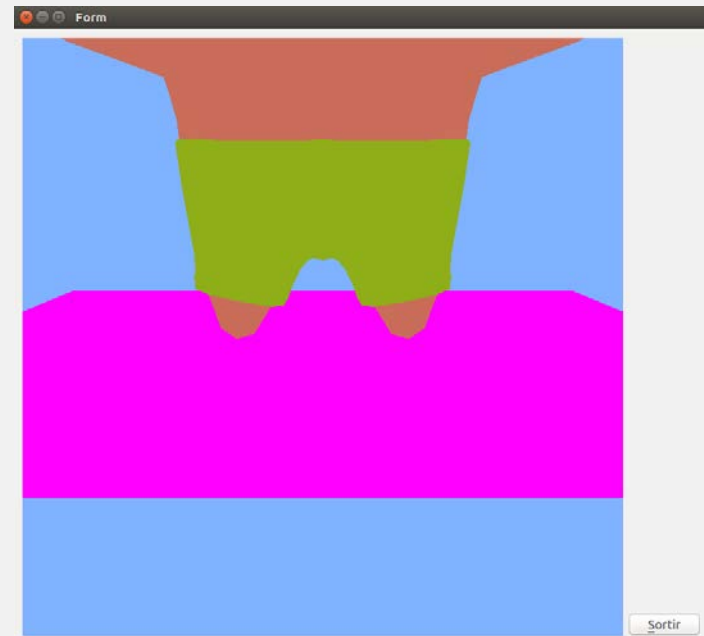
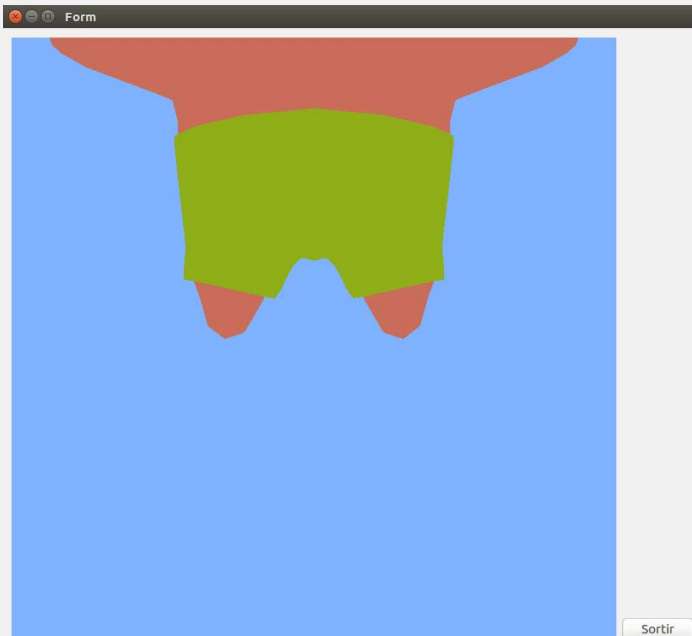
# Pintar objecte qualsevol

## (exercici 4)

- Modifiquem el terra
  - Mida 5x5 i centrat a l'origen de coordenades
  - Canviem directament les coordenades dels vèrtexs

Amb **obs-vrp** paral·lel a Z i **vrp.y=0**

Amb **obs-vrp** no paral·lel a Z

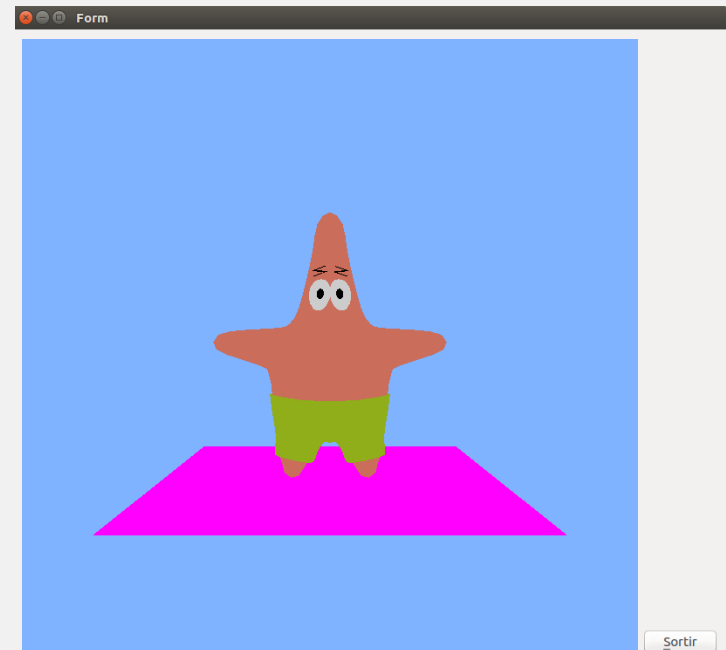
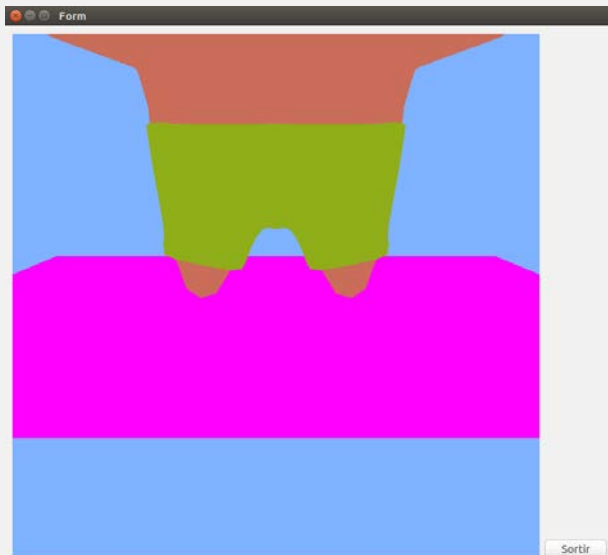


# Pintar objecte qualsevol

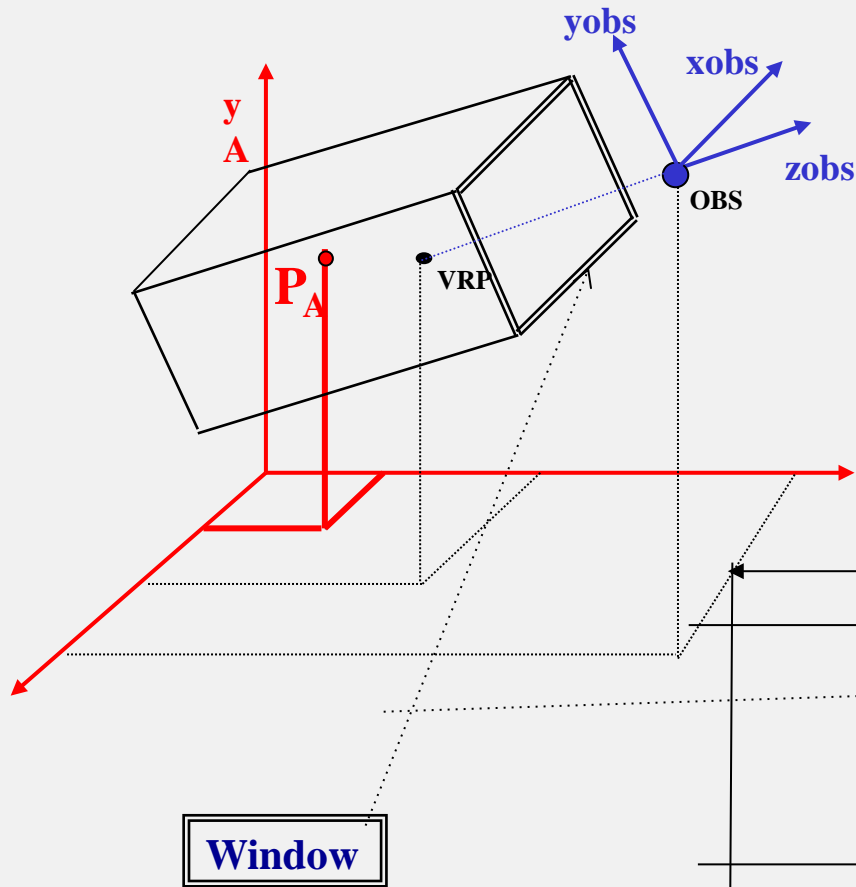
## (exercici 4)

- Recalculem càmera
  - Patricio i terra no hi caben a la càmera que tenim
  - Cal recalcular els paràmetres (de posició i orientació i òptica) de la càmera perspectiva per a veure'l sencer i ocupant el màxim del *viewport*

*la capsa de l'escena es calcula a partir e les dades del terra i del Patricio, que són conegudes.*

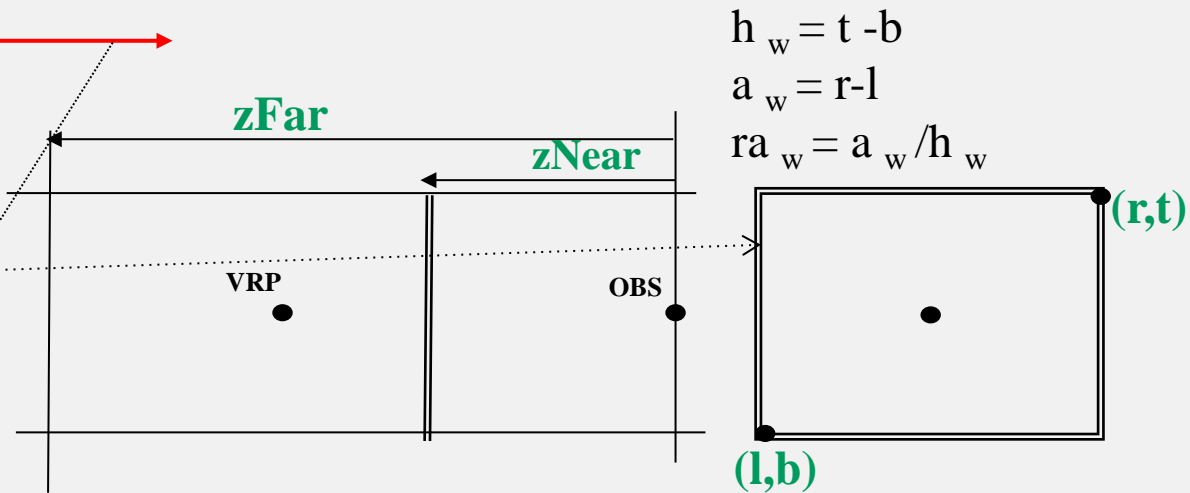


# Càmera ortogonal (exercici 5)



**Tipus de càmera: Axonomètrica:**

$l, r, b, t, -window-, zNear, zFar$



# Càmera ortogonal (exercici 5)

- Càlcul matriu de projecció (òptica de la càmera) amb la crida:

```
glm::mat4 Proj = glm::ortho (left, right, bottom, top, ZNear, ZFar)
```

- Afegir la possibilitat de tenir les dues òptiques possibles i decidibles amb la tecla 'O':
  - Inicialment tenim òptica perspectiva i canviarem d'òptica cada cop que l'usuari premi la tecla 'O'

# *Resize* per a càmera ortogonal (exercici 6)

Afegir/modificar al mètode `resizeGL` el necessari per a que no deformi ni retalli tampoc amb aquesta òptica.

En un exemple on  $R$  és el radi de l'esfera tenim:

- Window mínim requerit (centrat) =  $(-R, R, -R, R) \Rightarrow$  una  $ra_w = 1$
- Si  $ra_w \neq ra_v \Rightarrow$  deformació
  - Si  $ra_v > 1 \Rightarrow$  cal incrementar la  $ra_w \Rightarrow$  *modificar window*  
com  $ra_w = a_w/h_w \Rightarrow$  podem incrementar  $a_w$  o decrementar  $h_w$  (és retallaria esfera!!)  
Per tant:  
 $a_w^* = ra_v * h_w = ra_v * 2R \Rightarrow inc\_a = a_w^* - a_w$   
 $window = (- (R + inc\_a/2), R + inc\_a/2, -R, R) = (-R \cdot ra_v, R \cdot ra_v, -R, R)$
  - raonament similar per recalculer window quan  $ra_v < 1$