

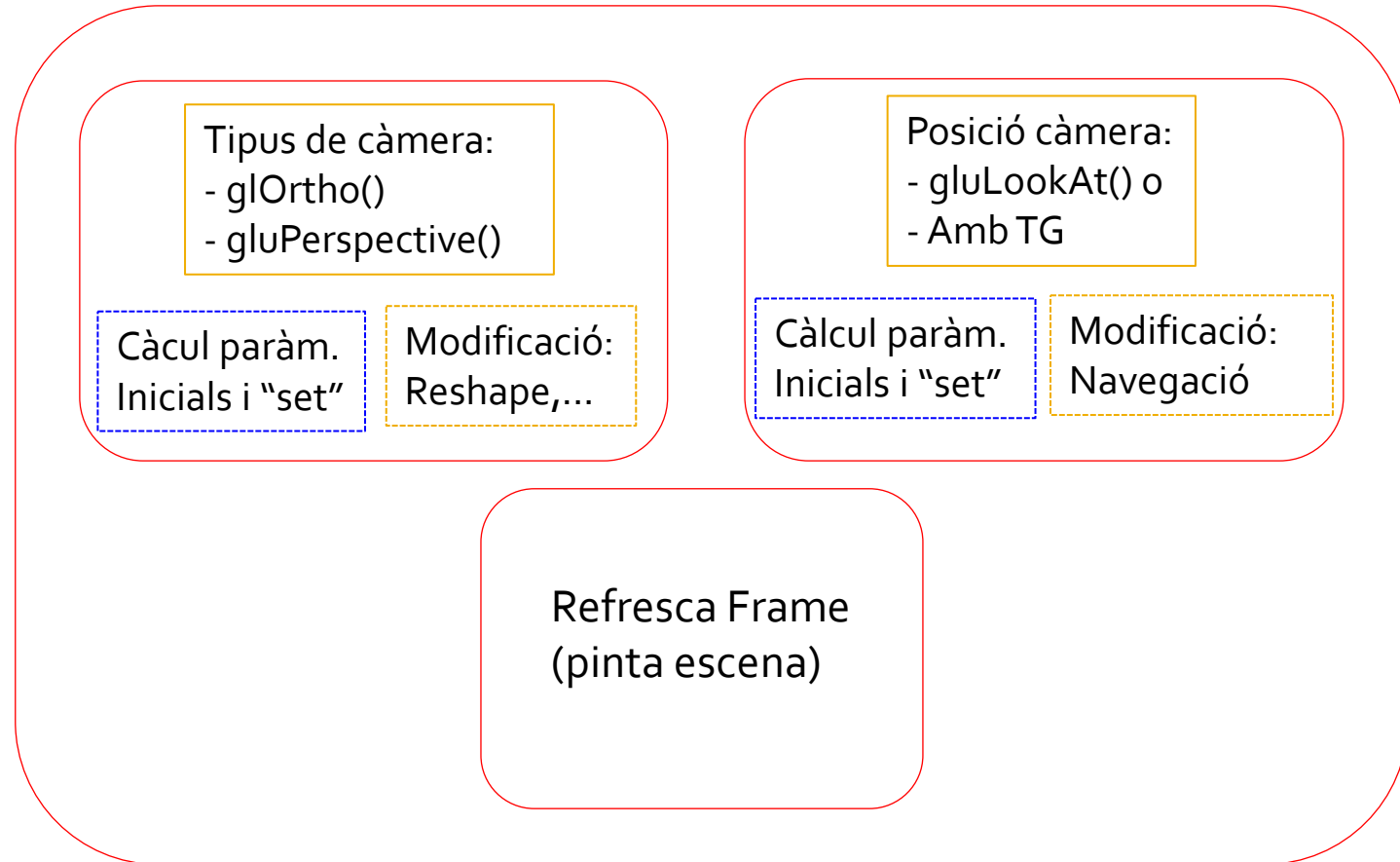
© Professors d'IDI – Curs 2014-2015

# Bloc\_3: Càmera. Sessió 1

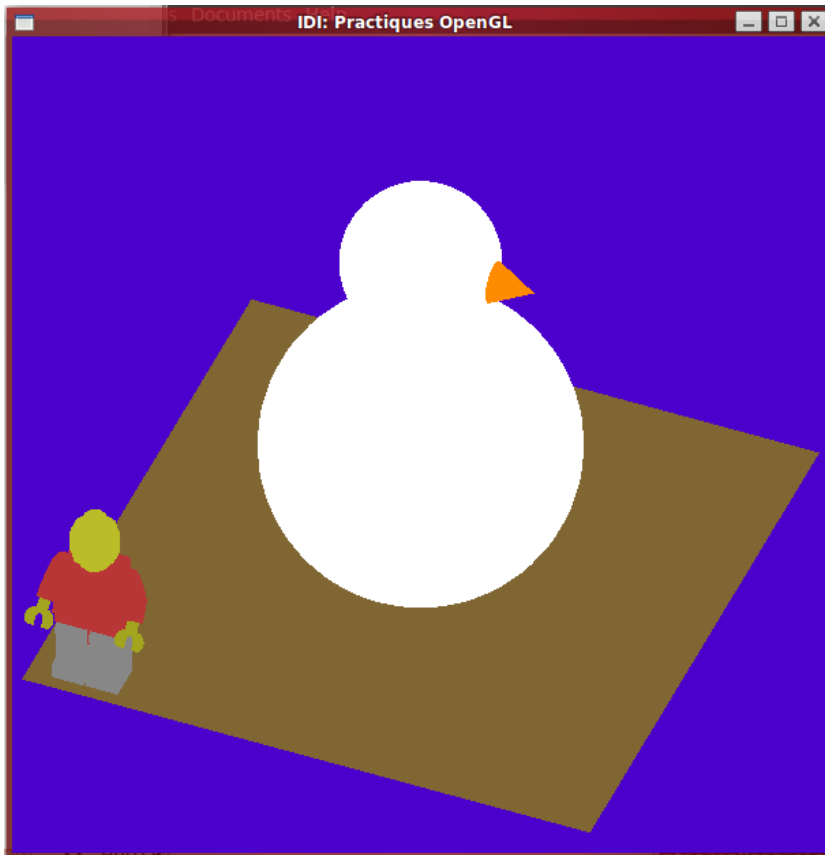
# Introducció

- **3 sessions** de laboratori.
- Llegiu el guió d'aquest Bloc.
- Objectiu global:
  - Inicialitzacions de càmera per complir els requeriments d'una aplicació
  - saber utilitzar les funcions d'OpenGL
  - estructurar aplicacions, i practicar tècniques de modificació interactiva de càmera.
- Sessió1: seccions 1 a 5 del guió.
- Sessió 2: seccions 6, 7 i 8.
- Sessió 3: secció 9 i aplicació a lliurar (secció 10).
- La secció 11 és totalment optativa amb funcionalitats més “avançades” que potser us interesen en algun moment...

# Estructura recomanda per a l'aplicació



# Escena de test per a la sessió 1



- Escena de l'aplicació que heu lliurat del Bloc 2.
- Netejeu el seu codi per a quedar-vos amb les funcionalitats demanades.
- Caldrà anar reorganitzant el codi amb les parts que us hem suggerit.

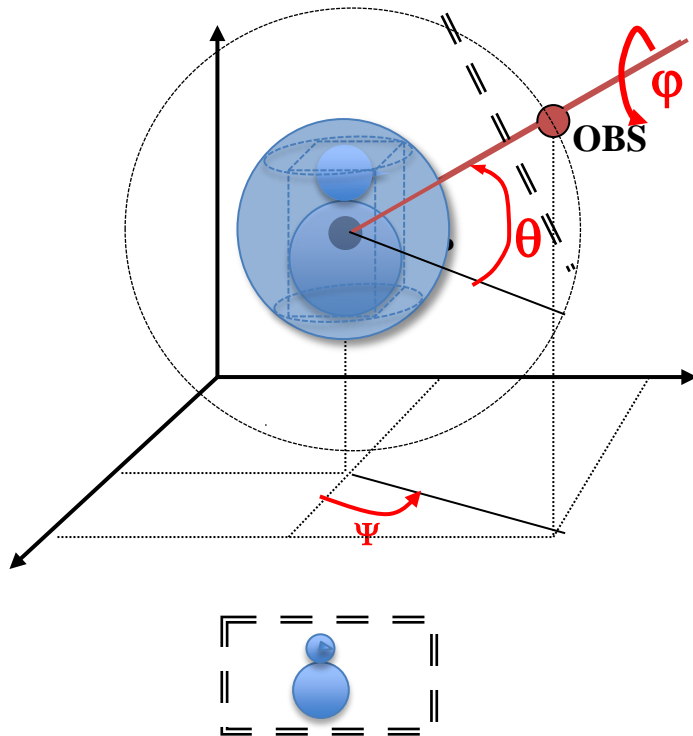
# Objectiu de la sessió 1



- Visualitzar l'escena:
  - L'escena s'ha de veure centrada i la posició de la càmera s'ha de declarar amb angles d'Euler (secció 3).
  - S'ha de poder inspeccionar l'escena interactivament amb el ratolí modificant els angles d'Euler (secció 3).
  - Sense retallar i sense deformació en un **viewport** que ocuparà sempre **tota la finestra gràfica** (secció 2).
  - L'òptica podrà ser axonomètrica o perspectiva (seleccionable per l'usuari) (seccions 2 i 4).

# Objectiu de la sessió 1: Repasseu l'exercici vist a classe de teoria

- Veure escena sempre sense retallar i sense deformació (en pas a viewport)
- La imatge inicial volem que estigui centrada i ocupant raonablement el viewport i mostri escena vista des d'una posició arbitrària.
- Permetre modificació interactiva de la càmera.



- 1) Càlcul de la capsula mínima contenidora **de tota l'escena**
  - Calcular vèrtex mínim i màxim
  - Calcular centre de la capsula
- 2) Càlcul de l'esfera (Radi)

# Càmera Axonomètrica. Secció 2

- `glOrtho` (left, right, bottom, top,  $zN, zF$ )
- Càlcul de la inicialització dels paràmetres per veure tota l'escena ocupant, raonablement, el volum de visió:
  - Utilitzeu les dades calculades de l'esfera contenidora.
  - Guardeu les dades del “*window*”,  $zN$  i  $zF$  en “*ini\_càmera*” (fixeu-vos que en càmera axonomètrica el  $zN$  pot ser negatiu –de fet, fins que tinguem posicionament de càmera, caldrà que sigui negatiu-).
- Nou Reshape per a no tenir deformacions, maximitzant l'ocupació de la finestra gràfica.
  - Viewport que ocupi tota la finestra gràfica.
  - Modificar *window* per a què la seva  $ra_w$  sigui igual a la  $ra_v$  del viewport. Sempre a partir dels seus valors inicials.

# Posicionament amb angles d'Euler.

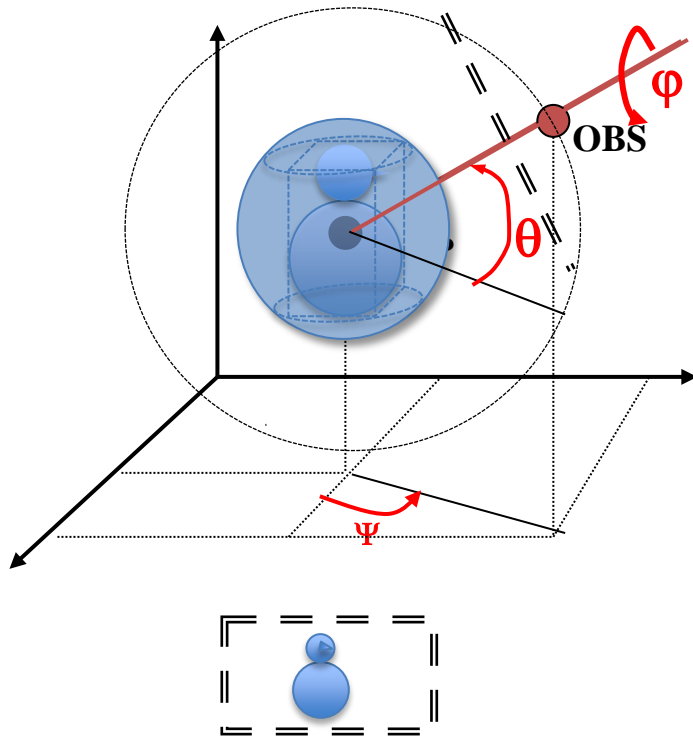
## Secció 3

- Ja tenim la capsula contenidora de l'escena, per tant podem:
  - Decidir el punt de referència de visió (**VRP**)
  - Decidir la distància **dist** de l'observador al VRP:  
 $dist > Radi$
  - Decidir els angles d'Euler inicials:  
seran arbitraris (però coneguts per vosaltres ☺ ).  
Possibilitat de modificar-los amb el ratolí (que ja ho teníeu), però recordeu que l'usuari ha de pensar que mou la càmera.
  - Recomanació: feu mètode específic "ini\_pos\_camera".
- **Modificar els paràmetres de zN i zF:**
  - Fixeu-vos que ara tenim tota l'escena per davant de l'observador, per tant als paràmetres de l'òptica de la càmera que teniu definits (glOrtho) us caldrà modificar els plans de retallat de forma adient (zN i zF).



# Posicionament amb angles d'Euler.

## Secció 3



Transformacions amb angles d'Euler:

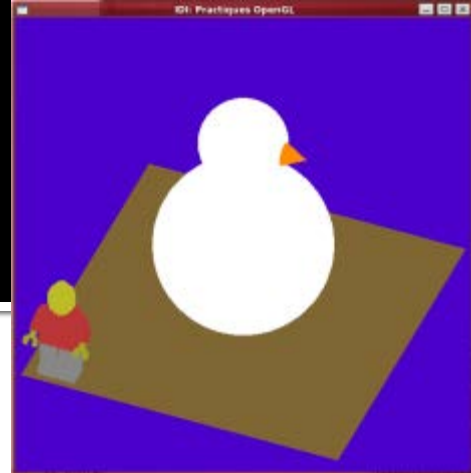
```
glMatrixMode (GL_MODELVIEW);  
glLoadIdentity ();  
glTranslatef (0., 0., -dist)  
glRotatef ( $\varphi$ , 0., 0., 1.)  
glRotatef ( $\theta$ , 1., 0., 0.)  
glRotatef ( $-\psi$ , 0., 1., 0.)  
glTranslatef (-VRP.x, -VRP.y, -VRP.z))
```

- Modifiquen interactivament els angles d'Euler amb el ratolí.  
Poseu valors màxims als dos angles d'Euler:  
 $\Psi$  en  $[0...360^\circ]$ ,  
 $\theta$  en  $[-90...90]$

# Càmera Perspectiva. Secció 4

- `gluPerspective` (FOV,  $ra_w$ ,  $zN$ ,  $zF$ )
- Càlcul de la inicialització dels paràmetres que permeti veure tota l'escena: sense retallar i maximitzant ocupació del volum de visió:
  - Feu els càlculs pensant en l'esfera englobant.
  - Calculeu i guardeu valors inicials de paràmetres.
- Nou Reshape per a no tenir deformacions, ocupant raonablement la finestra gràfica.
  - Viewport que ocupi tota la finestra gràfica
  - **Feu sempre els càlculs a partir del valor inicial calculat de FOV.**
- Tecla que modifica el tipus de càmera actiu: 'p' (axonomètrica vs. perspectiva).
- Afegiu `include GL/glu.h` i en `make` la llibreria `-IGLU`.
- Analitzeu les diferències “visuals” amb la càmera axonomètrica.

# Exercici 1. Secció 5



- Escena final del bloc 2.
- Inicialment es mostra l'escena des d'una posició arbitrària, sense retallar, amb una càmera perspectiva i aprofitant al màxim la grandària del *viewport*, sense deformació.
- Posició de la càmera definida amb angles d'Euler.
- Amb la tecla 'p' podem alternar entre càmera perspectiva i axonomètrica. En cap cas ha d'haver-hi deformació si l'usuari fa un *resize* de la finestra gràfica.
- Permet inspeccionar l'escena modificant els angles d'Euler amb el ratolí.
- Permet fer un *reset* per a tornar a la visualització de l'escena des de la càmera inicial.
- Amb la tecla 'h' es mostra un *help* que indica com activar les diferents funcionalitats.