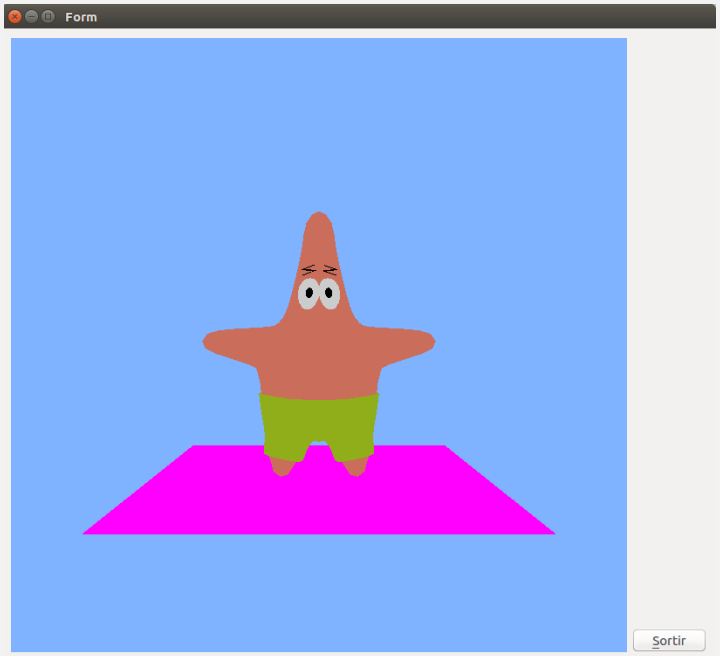
Laboratori INDI

Sessió 2.3

Punt de partida

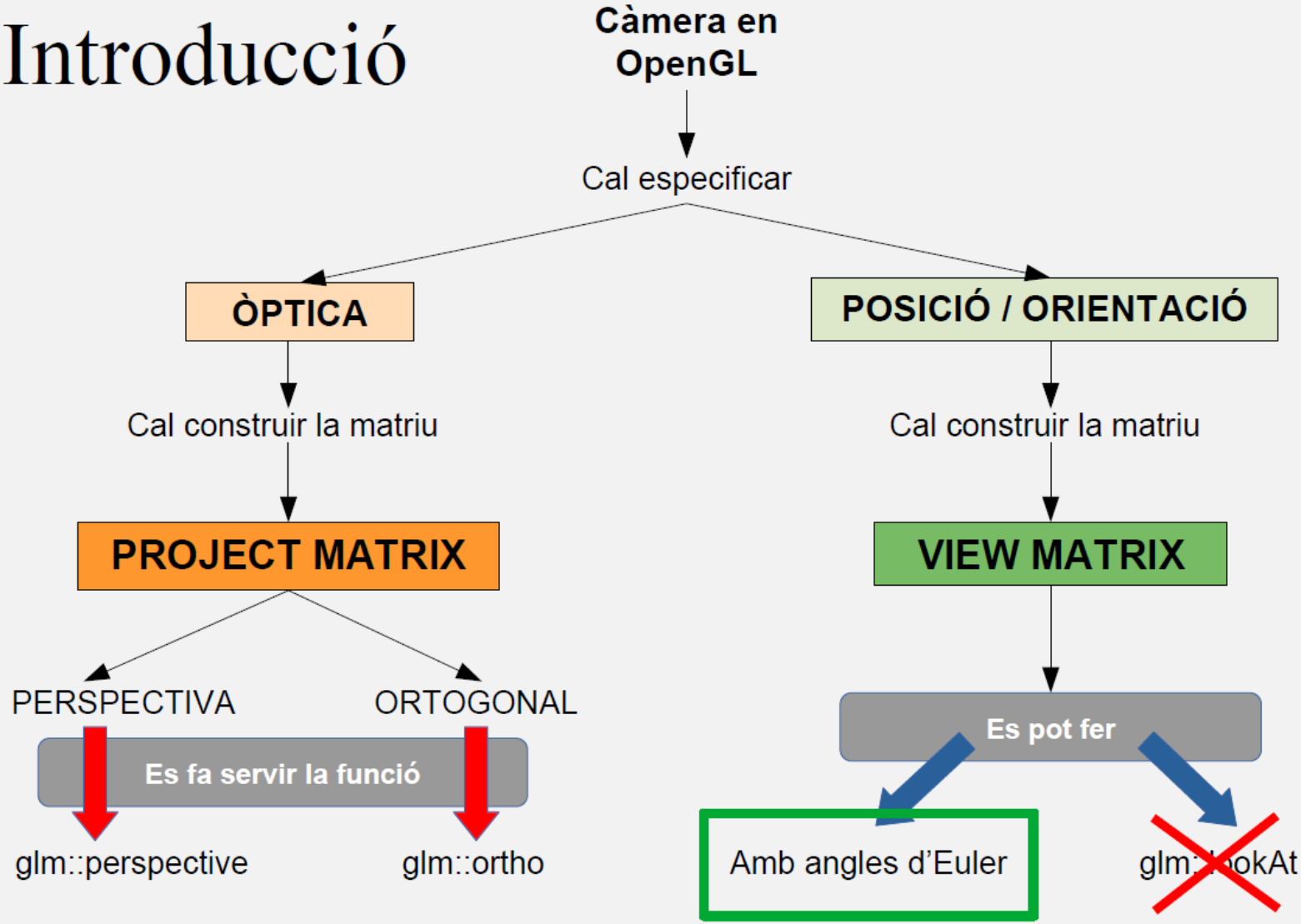
- Escena: Terra de 5x5 centrat en (0,0,0) i en pla XZ + Patricio amb centre base de la seva capsa a l'origen i alçada 4
- Càmera en tercera persona amb òptica perspectiva i ortogonal



Laboratori OpenGL – Sessió 2.3

1. View Matrix amb angles d'Euler
2. Interacció per inspecció (amb angles d'Euler)
3. Zoom (òptica perspectiva i ortogonal -opcional-)
4. Creació d'una escena més complexa

Introducció



View Matrix amb angles d'Euler

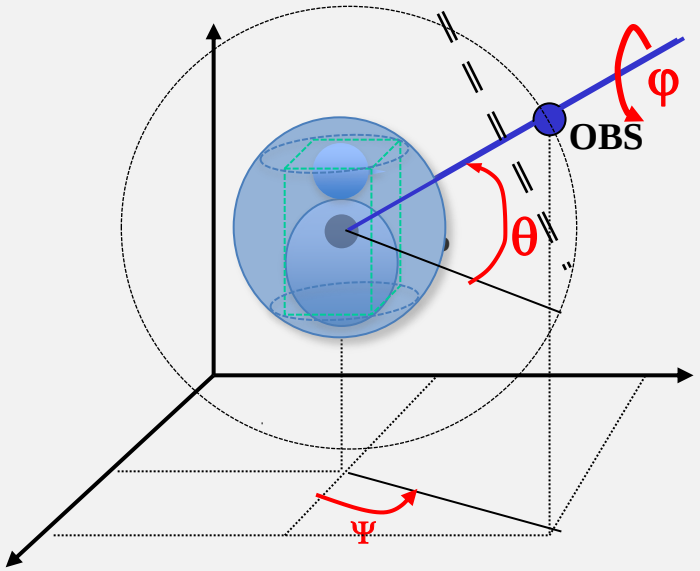
- LookAt és molt intuïtiu.
- No és fàcil fer girs de l'observador al voltant de l'escena amb lookAt.
- Usarem angles d'Euler que són com les coordenades polars de la terra.
- És més fàcil fer que l'observador roti al voltant de l'escena.

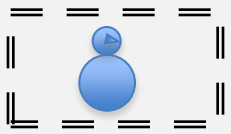
View Matrix amb angles d'Euler

- Fins ara: lookAt per col·locar la càmera.
- En aquesta sessió HEU D'ELIMINAR lookAt.
- Construireu la matriu View utilitzant TG.
- Paràmetres que calen:
  - VRP
  - θ : Angle de gir respecte Y
  - Ψ : Angle de gir respecte X
  - d (Distància entre OBS i VRP)
- No ens cal l'OBS. Amb angles d'Euler, l'observador està implícit.

Transf. *view* amb angles d'Euler

(exercici 1)





```
VM=Translate (0.,0.,-d)
VM=VM*Rotate(-φ,0,0,1)
VM= VM*Rotate (θ,1.,0.,0.)
VM= VM*Rotate(-ψ.,0.,1.,0.)
VM= VM*Translate(-VRP.x,-VRP.y,-VRP.z)
viewMatrix(VM)
```

Atenció a l'ordre!

ψ angle de gir respecte Y
θ angle de gir respecte X

Angles donats en RADIANS!

Compte amb signes:

• Si s'ha calculat ψ positiu quan càmera gira cap a la dreta, serà un gir anti-horari respecte eix Y de la càmera, per tant, matemàticament positiu; com girem els objectes en sentit contrari, cal posar −ψ en el codi.

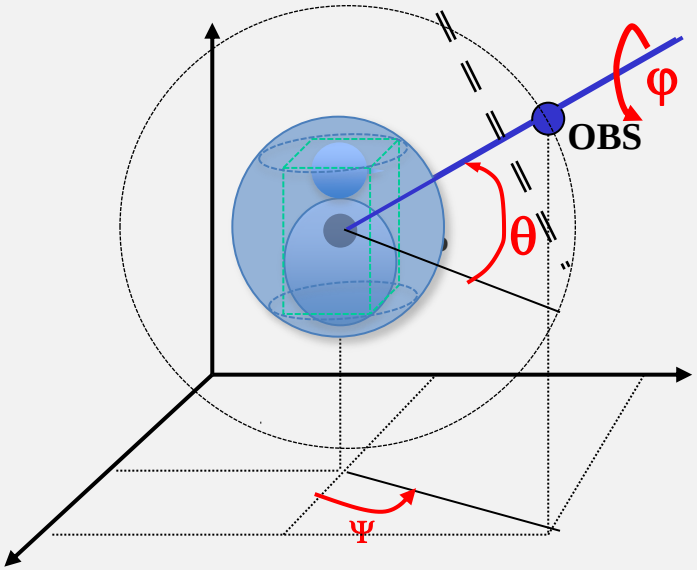
•Si s'ha calculat θ positiu quan pugem la càmera, serà un gir horari; per tant, matemàticament un gir negatiu; com objecte girarà en sentit contrari (anti-horari), ja és correcte deixar signe positu.

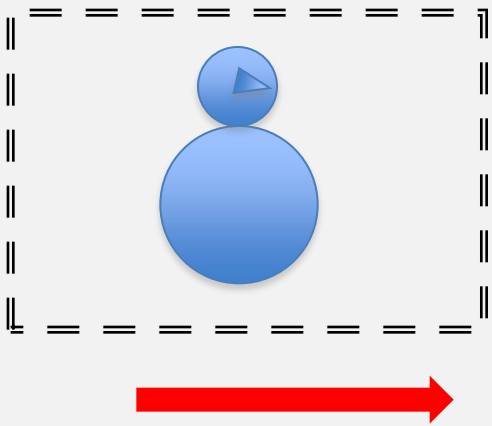
Laboratori OpenGL – Sessió 2.3

1. View Matrix amb angles d'Euler
2. Interacció per inspecció (amb angles d'Euler)
3. Zoom (òptica perspectiva i ortogonal -opcional-)
4. Creació d'una escena més complexa

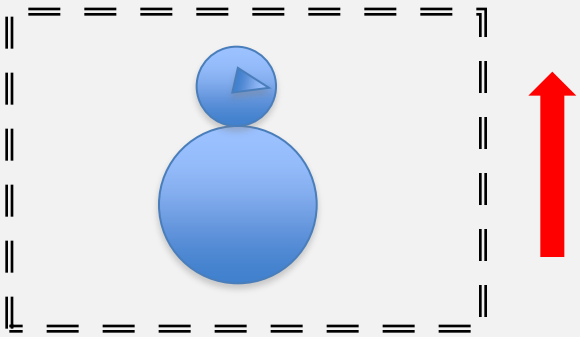
Interacció amb angles d'Euler

(exercici 2)





Moviment del ratolí d'esquerra a dreta ➔ increment angle Ψ



Moviment del ratolí de baix a dalt ➔ increment angle θ

Interacció amb el ratolí (1)

- Per tal de tractar events de baix nivell en una aplicació OpenGL amb Qt cal re-implementar els mètodes virtuals corresponents (a la classe MyGLWidget):

  virtual void mousePressEvent (QMouseEvent * e)

  virtual void mouseReleaseEvent (QMouseEvent * e)

  virtual void mouseMoveEvent (QMouseEvent * e)

  virtual void keyPressEvent (QKeyEvent * e)

  - En MyGLWidget.h caldrà afegir: #include <QMouseEvent>
    i declarar el mètodes virtuals

Interacció amb el ratolí (2)

- La funció mouseMoveEvent s'executa quan es mou el ratolí.
- Aquesta funció rep un paràmetre:

```
QmouseEvent *e
```

- e→x() i e→y() retornen la posició del ratolí en pantalla.

Interacció amb el ratolí (3)

- Per saber els graus necessaris pels angles d'Euler es pot guardar la posició anterior i fer una resta:

```
angleX = e→x() - xAnt

angleY = e→y() - yAnt
```

- Això ens dóna un nombre de píxels que caldrà transformar a radians.
- Cal actualitzar `xAnt` i `yAnt` cada vegada que es mou el ratolí.

Interacció amb el ratolí (4)

Es vol que el moviment de càmera es faci prement el **botó esquerre** del ratolí, i no qualsevol.

- Si volem controlar el botó del ratolí que s'usa:

```
if ( e->buttons() == Qt::LeftButton )  // e és QMouseEvent
```

- Si volem controlar que a més no s'ha usat cap modificador (Shift, Ctrl, Alt):

```
if ( e->buttons() == Qt::LeftButton &&
     ! ( e->modifiers() &
         ( Qt::ShiftModifier | Qt::AltModifier | Qt::ControlModifier ) ) )
// controla que s'ha premut botó esquerre i cap modificador
```

Interacció amb el ratolí (5)

- El ratolí controlarà 2 moviments: rotació i zoom.
- Utilitzeu una variable d'estat, `CurrentAction,` que ens diferenciï si estem en mode rotació, zoom o no fem res:

```
typedef enum {ROTATE, NONE, ZOOM} Action;
Action CurrentAction;
```

- A `mousePressEvent` asigneu a `CurrentAction` el mode corresponent.

Laboratori OpenGL – Sessió 2.3

1. View Matrix amb angles d'Euler
2. Interacció per inspecció (amb angles d'Euler)
3. Zoom (òptica perspectiva i ortogonal -opcional-)
4. Creació d'una escena més complexa

Zoom
(exercici 3)

- Per a fer un zoom ho farem modificant l'angle d'obertura de la càmera (FOV)
  - Zoom-in → decrementar l'angle FOV (tecla 'Z')
  - Zoom-out → incrementar l'angle FOV (tecla 'X')
- La quantitat en què incrementem o decrementem el FOV és fixa.
- L'angle d'obertura **ha ser** menor que 180º i major que 0º. Sinó podem tenir efectes estranys.

Zoom
(exercici 3)

- El zoom i el resize s'han de combinar correctament (sense salts).
- Si feu zoom i després resize, el zoom no s'ha de perdre.
- Per tal que es conservi el que cal fer és:

$$\alpha_{inicial} = FOV_{zoom} / 2$$

- L'angle inicial ha de ser l'angle que queda després de modificar FOV per fer zoom.

Zoom
(exercici 3)

- [OPCIONAL] Implementar el zoom per la càmera ortogonal. Caldria que modifiquéssiu la mida de la window (left, right, bottom, top) mantenint la relació d'aspecte.

- [OPCIONAL] Fer que l'usuari també pugui fer zoom de l'escena prement SHIFT + botó esquerre del ratolí + moviment del ratolí endavant o enrere.

Laboratori OpenGL – Sessió 2.3

1. View Matrix amb angles d'Euler
2. Interacció per inspecció (amb angles d'Euler)
3. Zoom (òptica perspectiva i ortogonal -opcional-)
4. Creació d'una escena més complexa

Escena completa (exercici 4)

Modifiqueu la vostra escena per a veure el que es veu a la imatge.

Nova escena formada per:

- Terra de 5x5 centrat al (0,0,0)
- Tres Patricios d'alçada 1 amb centres base en (2,0,2), (0,0,0) i (-2,0,-2). El primer direcció Z+, el segon direcció X+ i el tercer direcció Z-

Calen paràmetres de càmera per a veure-ho tot (3ª persona)

ATTENTION !

No carregeu el Patricio 3 cops !!

Recordatori

1. pMin i pMax de l'escena calculats "manualment"
2. La distància, d, triada per vosaltres. Típicament 2R.
3. El VRP és el centre de l'esfera embolcallant de l'**escena**, i no el centre de la capsa mínima contenidora del Patrício.
4. Trieu angles Eulers inicials al vostre gust.
5. No oblideu cridar makeCurrent() a l'inici de cada funció de gestió d'esdeveniments.