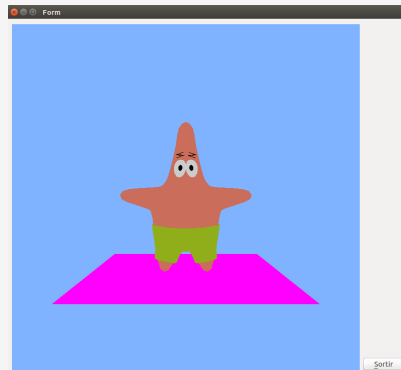


Escena i funcionalitats de partida

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



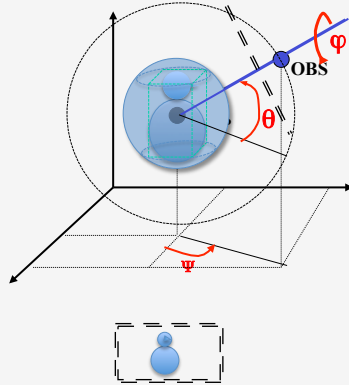
IDI Q2 2018-2019

Laboratori OpenGL – Sessió 2.3

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

IDI Q2 2018-2019

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!

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 $-\psi$ 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 positiu.

IDI Q2 2018-2019

Consells d'estructuració de codi (1)

- En **ini_càmera** inicialitzeu **TAMBE** angles i VRP i d si no els teniu ja inicialitzats per atènyer mateixa vista inicial de l'escena de la sessió 2.
- Feu una NOVA `viewTransform()`

```
void MyGLWidget::viewTransform()
{
    glm::mat4 VM(1.0f);

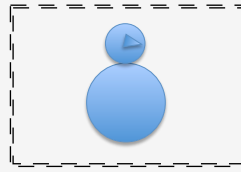
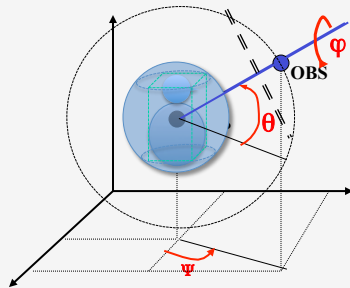
    //aquí codi de transformacions

    glUniformMatrix4fv(viewLoc, 1, GL_FALSE, &VM[0][0]);
}
```

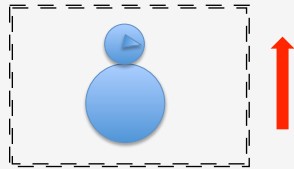
IDI Q2 2018-2019

Interacció amb angles d'Euler (1)

(exercici 2)



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



Moviment del ratolí de baix a dalt → increment angle θ

IDI Q2 2018-2019

Interacció directa amb Qt (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: `#include <QMouseEvent>`
i declarar el mètodes virtuals

IDI Q2 2018-2019

Interacció directa amb Qt (2)

- Exemple d'implementació:

```
void MyGLWidget::mousePressEvent (QMouseEvent *e)
{ makeCurrent();
  // posició actual ratolí per Qt
  xClick = e->x();
  yClick = e->y();
  /*si podem tenir diferents "estats" d'interacció,
  determinar l'estat segons botó premut */
  if (e->button() == Qt::LeftButton )
  {
    DoingInteractive = ROTATE;
  }
}
```

Si podem tenir diferents “estats” d’interacció, caldrà haver declarat en .h, per exemple:

```
typedef enum {ROTATE, NONE, ZOOM} InteractiveAction;
InteractiveAction DoingInteractive;
```

IDI Q2 2018-2019

Interacció directa amb Qt

```
void MyGLWidget::mousePressEvent ( QMouseEvent * e )
```

```
....
if ( e->button() == Qt::LeftButton )
    DoingInteractive=ROTATE;
if ( e->button() == Qt::RightButton )
    DoingInteractive=ZOOM;
```

IDI Q2 2018-2019

Interacció directa amb Qt (3)

- Exemple d'implementació:

```
void MyGLWidget::mouseMoveEvent (QMouseEvent *e)
{
    makecurrent();
    if (DoingInteraction==ROTATE) {
        psi+=(e->x()-xClick)*M_PI/180.0;
        ...
        xclick=e->x();
        viewTransform();
    }
    update();
}
```

IDI Q2 2018-2019

Interacció directa amb Qt (4)

- Exemple d'implementació:

```
void MyGLWidget::mouseReleaseEvent (QMouseEvent *e)
{
    makecurrent();
    DoingInteraction=NONE;
}
```

IDI Q2 2018-2019

Consells d'interacció amb ratolí

- No oblideu cridar a `makeCurrent()` sempre que entreu en un mètode de Qt que tracta un event
- 1) `mousePressEvent(...)`
 - Identifiqueu posició inicial de ratolí
 - Identifiqueu el botó premut => identificar l'acció d'interacció al moure el cursor (**inspecció**, escalat, zoom,...)
- 2) `mouseMoveEvent ()`
 - Recollir posició cursor; càlcul increment de posició
 - Segons tipus d'interacció: **actualitzar angles**, escalat, FOV,...
- 3) `mouseReleaseEvent()`
 - No volem cap tipus d'interacció al moure el ratolí

IDI Q2 2018-2019

Interacció amb angles d'Euler (2) (exercici 2)

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`

IDI Q2 2018-2019

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')

Noteu que heu de controlar el valor màxim i mínim del FOV

- Opcional: També ho podeu fer amb el ratolí i el botó dret
- Per a càmera ortogonal (opcional):
 - Modificar el window (left, right, bottom, top) mantenint ra

IDI Q2 2018-2019

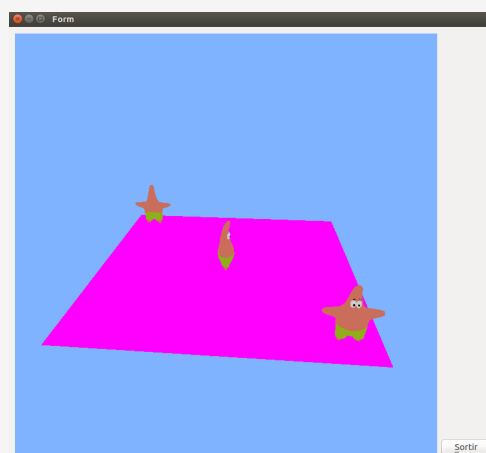
Escena completa (exercici 4)

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)



IDI Q2 2018-2019

Consells d'estructuració de codi

- Crear en *createBuffers()* un VAO per cada model (els tres patricios tenen el mateix VAO (caldrà carregar el model)).
- Tenir mètode *CalculCapsaModelPatricio()*
calcula la capsa d'un model i els paràmetres d'escalat, rotació i posició requerits per a calcular la TG que el posa a la posició requerida de l'escena
- Un mètode *ModelTransform()* per crear la TG de cada model.
- Mètode *IniEsfera()* en el que en funció de les dades de l'escena que tenim, inicialitzem la seva capsa i es calcula centre i radi esfera.
- Mètode *IniCàmera()* que inicialitza paràmetres càmera inicial.
- En *initializeGL()* cridar a iniEsfera, iniCamera i viewTransform (sempre tot després d'haver creat VAOs i compilat els shaders)
- Modificar *PainGL()*

IDI Q2 2018-2019