void MyGLWidget::viewTransform ()

{

glm::mat4 View; // Matriu de posició i orientació

View = glm::lookAt(OBS, VRP, glm::vec3(0,1,0));

glUniformMatrix4fv (viewLoc, 1, GL\_FALSE, &View[0][0]);

}

Obs es pot substituir per glm::vec3(-1,1,-1)

També sha de modificar el projecttransform per si es vol canviar el FOV

Proj = glm::perspective(FOV2, s ra, 0.1f, 500.f);

### LIMITE DE ZOOM → WHEEL ZOOMING

void MyGLWidget::wheelEvent ( QWheelEvent \* event )

{

makeCurrent();

float f = event->delta()/120;

if (((FOV + f/15) > 0) && ((FOV + f/15) < float(M\_PI))) FOV += f/15;

projectTransform();

update ();

}

al .h: (protected) virtual void wheelEvent ( QWheelEvent \* event );

## 

## 

## 

## 

## PASSAR POSFOCUS AL VERTEX O FRAGMENT

declarar al carregashader()

posfocusLoc = glGetUniformLocation (program->programId(), "posfocus");

i fer la funcio ini focus

void MyGLWidget::ini\_focus()

{

glm::vec3 posfocus(eix\_x, 4, eix\_z);

glUniform3fv (posfocusLoc, 1, &posfocus);

}

ZOOM i emit zoom

**NOTA IMPORTANT AL QTDESIGNER ASIGNA MINIM 2 MAXIMUM 179**

void MyGLWidget::zoomSlider(int zoom){

makeCurrent();

FOV = (M\_PI/180.0)\*zoom;

projectTransform();

update();

}

emit zoomSliderInverse(FOV\*180/M\_PI);

ROTAR i emit rotar

**NOTA IMPORTANT AL QTDESIGNER ASIGNA MINIM 2 MAXIMUM 179**

void MyGLWidget::rotarPatr(int rotarN){

makeCurrent();

rotar = (M\_PI/90.0)\*rotarN;

update();

}

emit updateDial(rotar\*90/M\_PI);