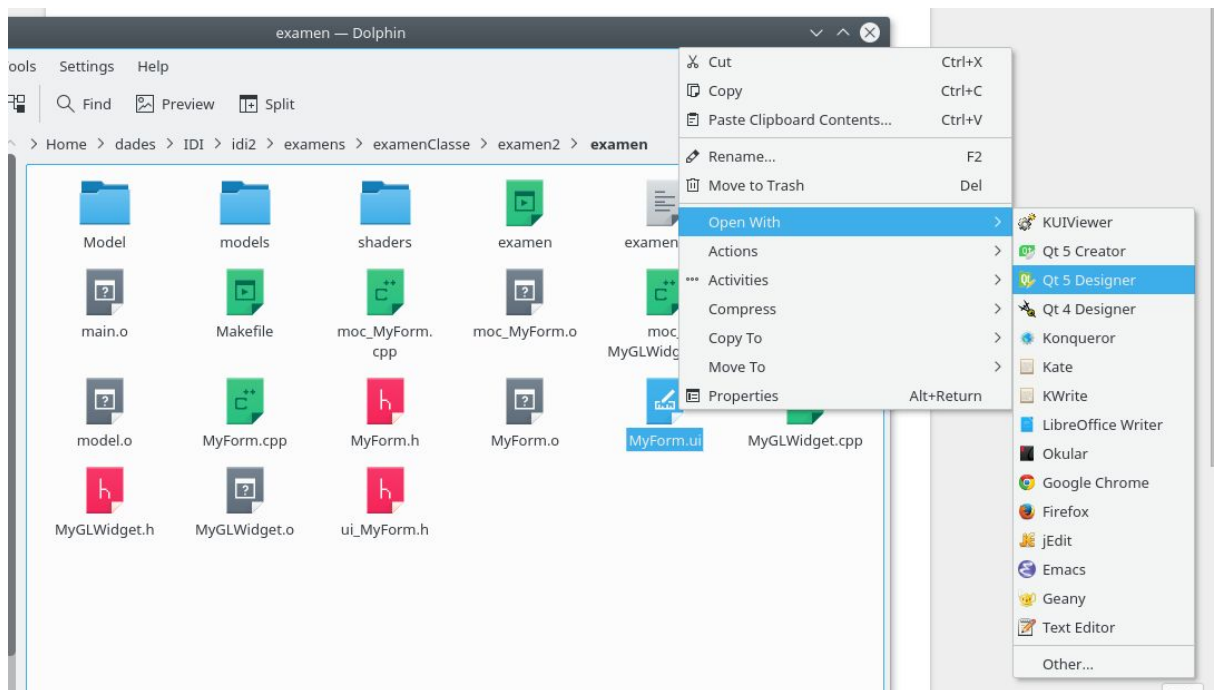
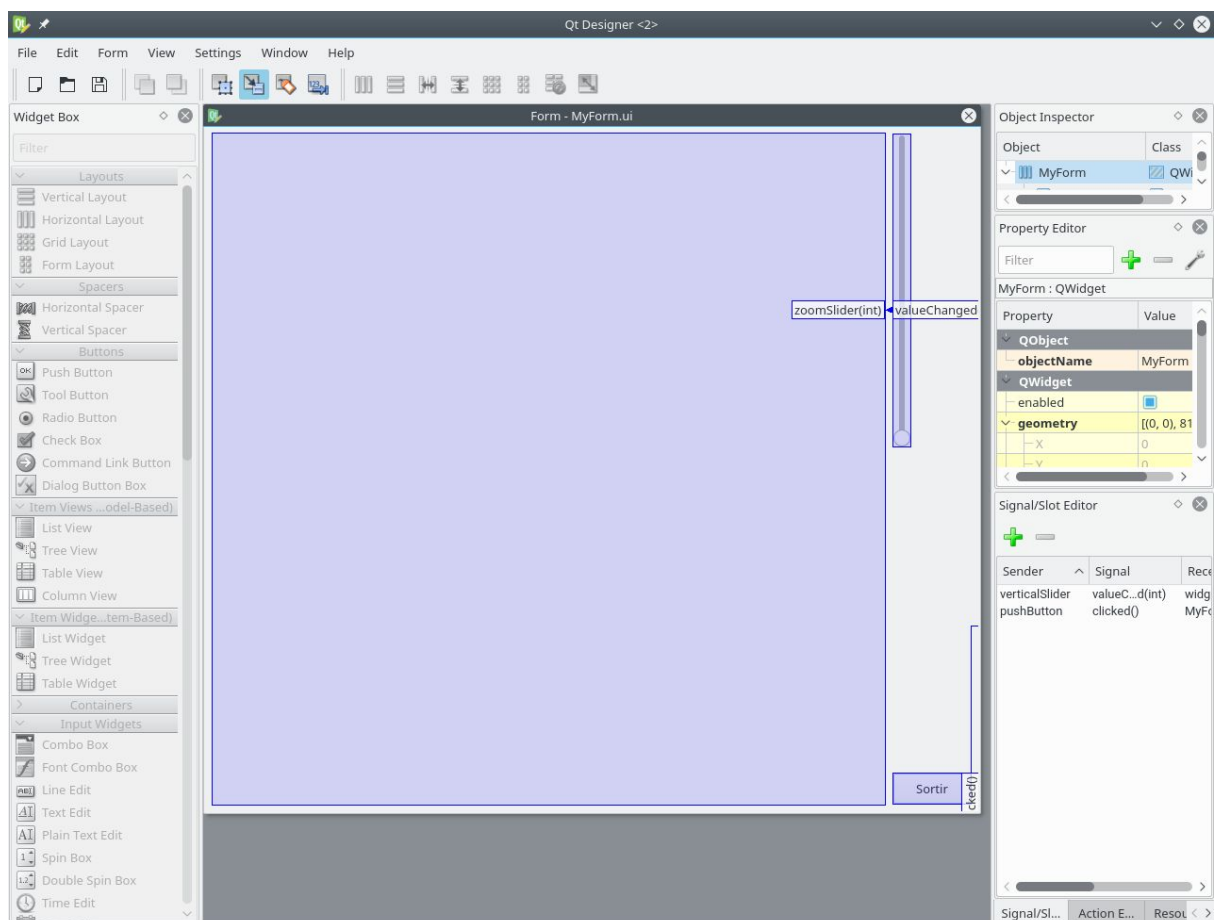


SLOTS (QT ---> WIDGET)

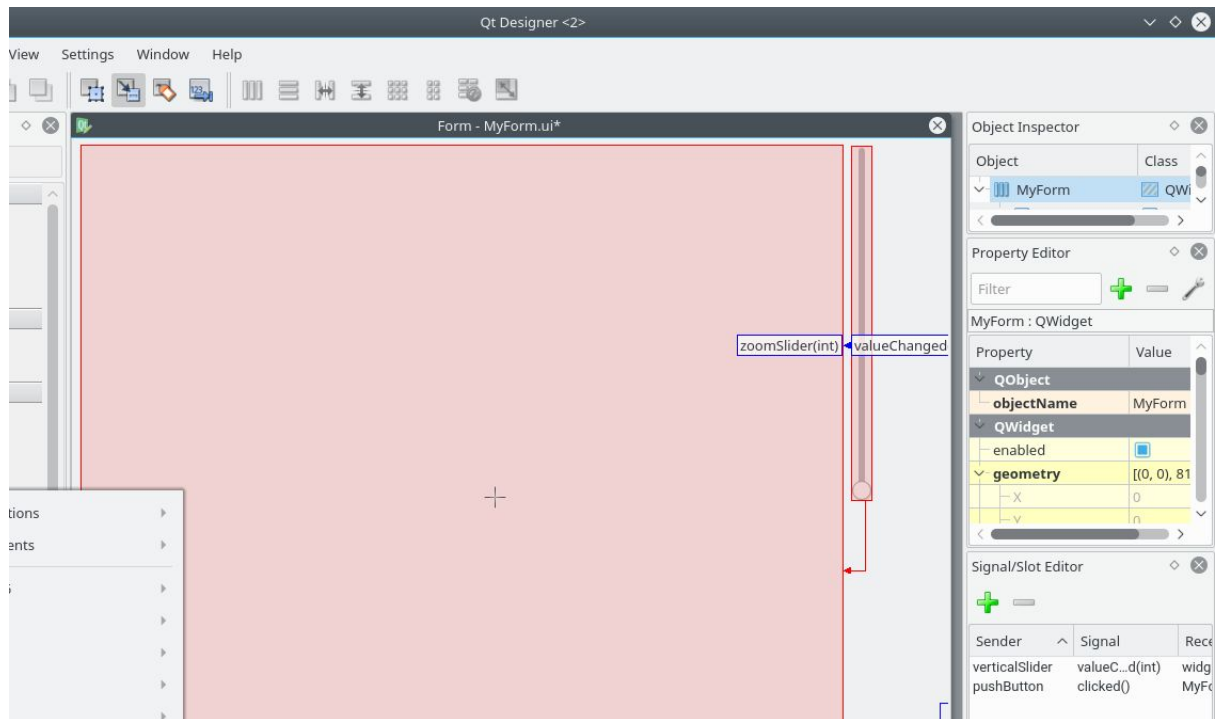
1. Obrir MyForm.ui amb Qt 5 Designer



2. Apretar el boto Edit Signals/Slots

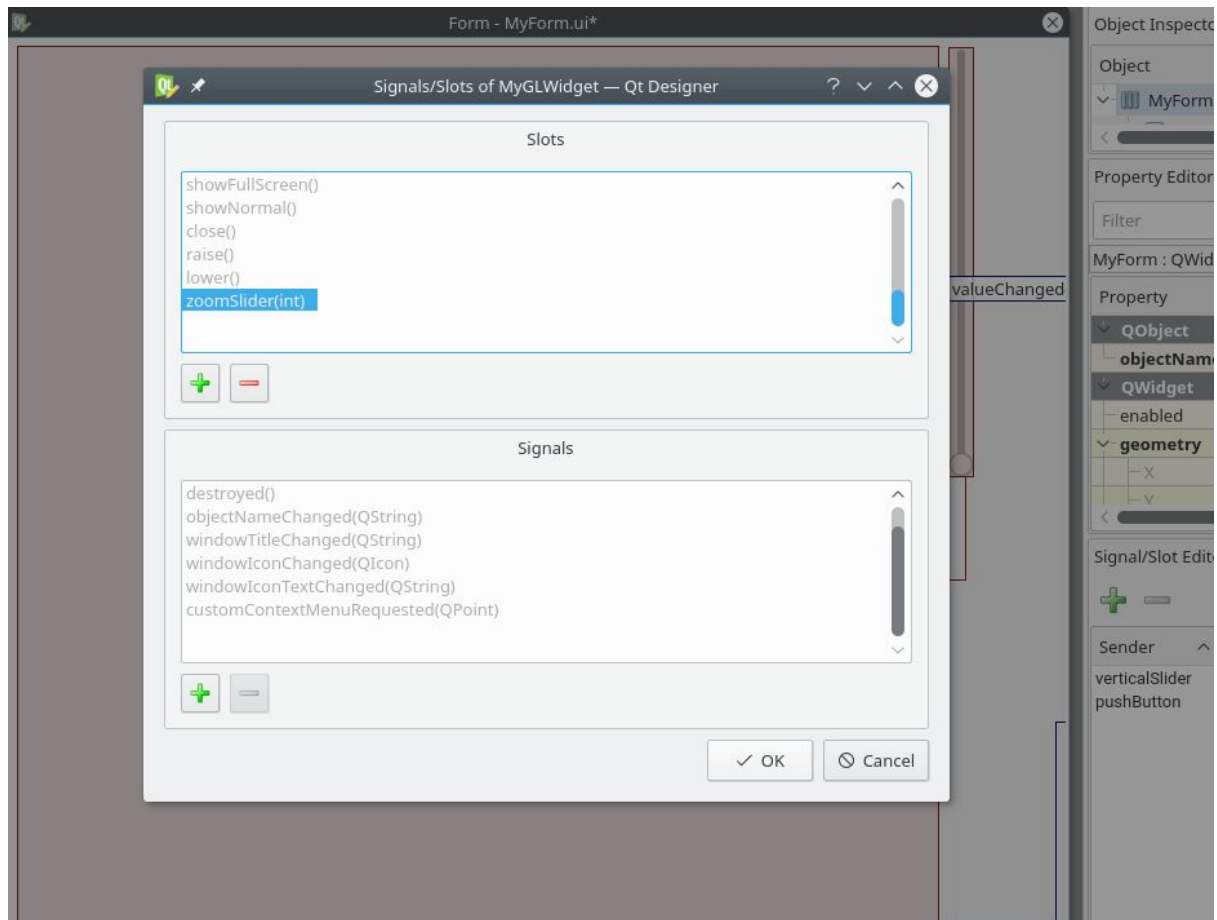


3. Arrostrar el slider a la pantalla al widget



4. Seleccionar valueChanged(int) (en el cas del slider) a la part esquerra. I per la part dreta, seleccionar "Edit".

A la part de Slots, ficar un nou amb el nom de la funció que vols i com a paràmetre, un int.
Per exemple: myFunction(int).



Seleccionar-la i prendre OK.

Al arxiu MyGLWidget.h afegir **abans de “public:”**

```
public slots:  
void myFunction(int value);
```

Al arxiu MyGLWidget.cpp afegir la funció corresponent:

```
void MyGLWidget::zoomSlider(int zoom){  
    makeCurrent();  
    codi;  
    projectTransform(); (o viewTransoform(), depenent de què ens demani l'exercici)  
    update();  
}
```

SIGNALS (WIDGET ----> QT)

Al codi

Declarar funció al .h (**signals**)

```
public slots:
    void zoomSlider(int value);

signals:
    void zoomSliderInverse(int value);
```

Afegir al codi: **emit zoomSliderInverse(valor que vols);**

S'ha d'afegir en el moment que el widget s'actualitza. Per exemple, en el cas del zoom, s'ha de posar al mateix lloc on actualitzes el FOV quan mous el ratolí.

```
if (DoingInteractive == ZOOM) {
    // Fem zoom
    FOV += (e->y() - yClick) * M_PI / 180.0;
    emit zoomSliderInverse(FOV*180/M_PI);

    projectTransform();
}
```

Exemple

```
case Qt::Key_P: { // canvia òptica entre perspectiva i axonomètrica
    patricio = !patricio;
    if(patricio)emit updateRadioButtonPatr();
    else emit updateRadioButtonPors();
    break;
}
```

AI Designer

Linkar (mirar més a dalt com hem fet als Slots). Aquest cop arrosseguem des del widget fins al Slider.

Però aquest cop, quan afegim el nom de la nostra funció, ho fem a la part de Signals, no Slots.