Laboratori:
Custom Widgets

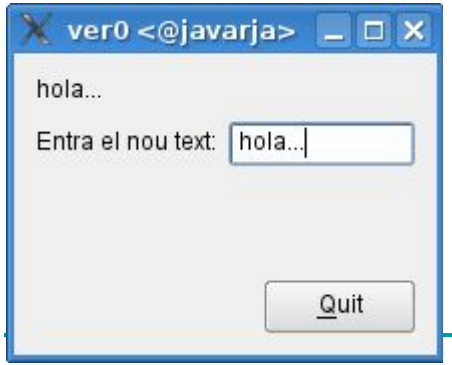
Professors de IDI

Index

- Qt: Recordatori
- Exemple 1: problema i solució
- Custom Widgets
- Promote
- Exemple 2: problema i solució
- MyGLWidget

Llibreria Qt: Recordatori

- Qt proporciona un conjunt de widgets configurables
- Es poden connectar mitjançant signals i slots
  - **Signal:** Esdeveniment que succeeix durant l'execució.
    - Ex: Clic sobre un widget...
  - **Slot:** mètode que s'executa en responsta a un signal

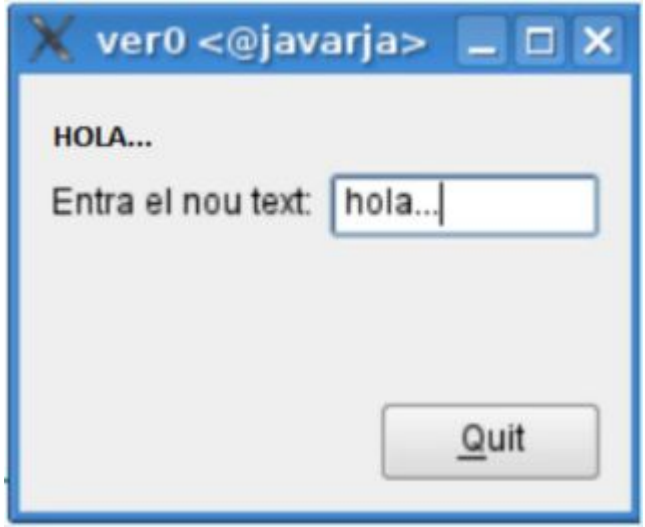


Exemple de signal i slot: escrivim al QLineEdit i apareix el text al QLabel

QLineEdit -> signal textChanged(QString)
QLabel -> slot setText(QString)

Llibreria Qt: Problema 1

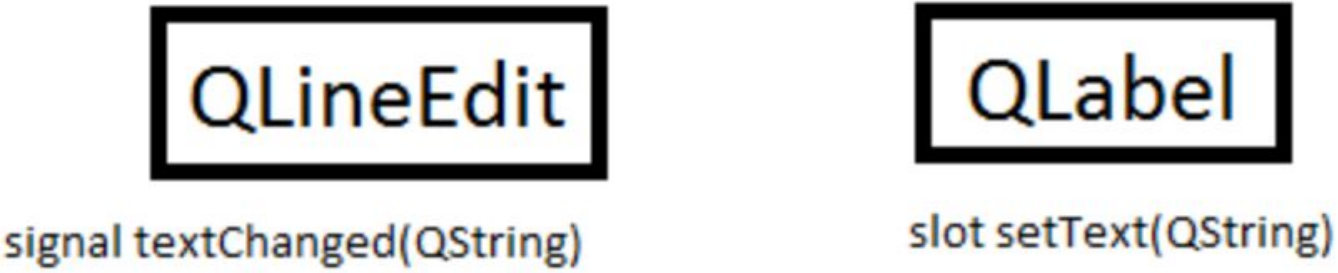
- Escrivim al QLineEdit i volem que el text aparegui en MAJÚSCULES al QLabel



- No hi ha cap slot a QLabel que converteixi a majúscules
- No podem accedir al codi del signal per canviar el text abans d'enviar-lo al QLabel
- No podem accedir al codi de l'slot per canviar el text
- QLabel i QLineEdit són classes
- Què fer ?

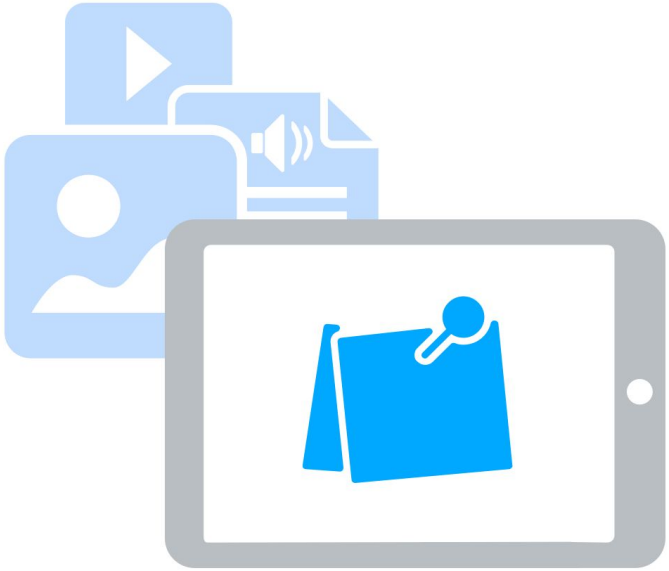
Llibreria Qt: Problema 1

- Com ho podem fer ?



- Cap de les dues classes es pot modificar
- Cap dels dos mètodes es pot modificar

Collaborate!



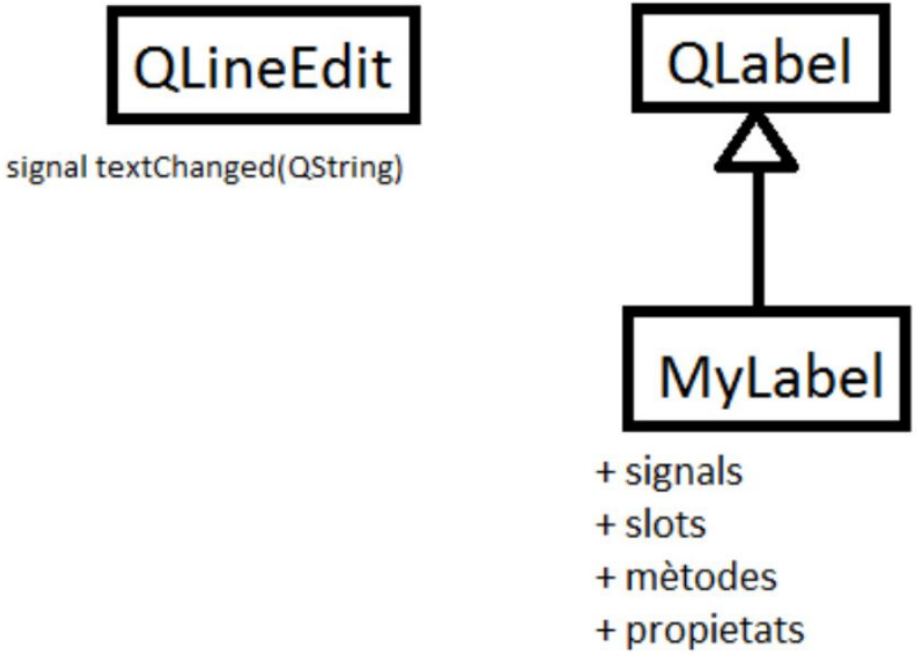
How to Edit

Click Edit This Slide in the plugin to make changes.

Don't have the Nearpod add-on? Open the "Add-ons" menu in Google Slides to install.

Llibreria Qt: Problema 1

- Com ho podem fer ? HEREDANT !



Llibreria Qt: Solució

- Heredar una classe nova de QLabel: MyLabel
- MyLabel hereda l'aspecte, mètodes i propietats, slots i signals de QLabel
- A MyLabel podem afegir nous slots i signals, si cal
- Podem escriure el codi dels slots però no dels signals
- Podem crear un slot que, abans de posar el text al QLabel, el converteixi a majúscules

```
void MyLabel::setTextMajuscules(QString s)
{
     QString str=s.toUpper();
     setText(str);      /// Mètode heredat de QLabel
}
```

- Després es pot connectar el signal textChanged(QString) a aquest slot
- Es pot heredar de qualsevol classe de Qt

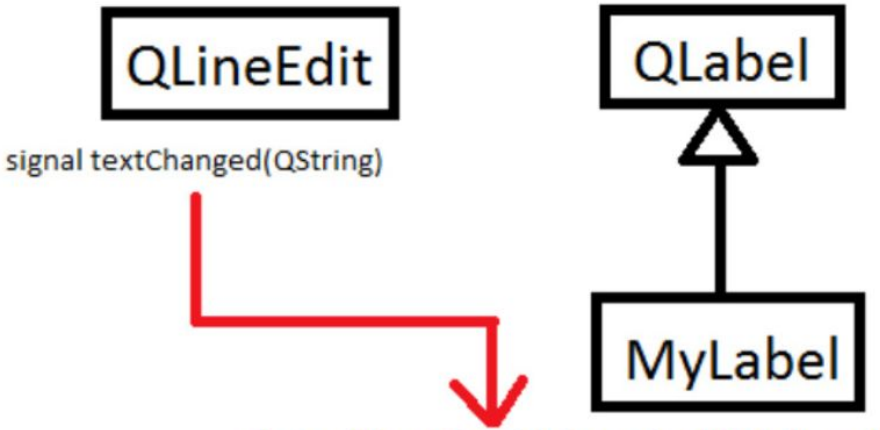
Llibreria Qt: Problema 1

- Com ho podem fer ?


```
slot void setTextMajuscules(QString s)
{
    QString str=s.toUpper();
    setText(str);  //HEREDAT !!!
}
```

Llibreria Qt: Problema 1

- Com ho podem fer ?



Llibreria Qt: Custom Widgets

MyLabel.h

```
#include <QLabel>
class MyLabel:public QLabel
{
     Q_OBJECT          // IMPORTANT !
     public:
          MyLabel(QWidget *parent);

     public slots:          // IMPORTANT !
          void setTextMajuscules(QString);
};     // No oblideu el ;
```

MyLabel.cpp

```
#include "MyLabel.h"

MyLabel::MyLabel(QWidget *parent=0):QLabel(parent)
{
}


void MyLabel::setTextMajuscules(QString s)
{
     QString str=s.toUpper();
     setText(str);     // heredat de QLabel
}
```

Llibreria Qt: Custom Widgets. Compilació

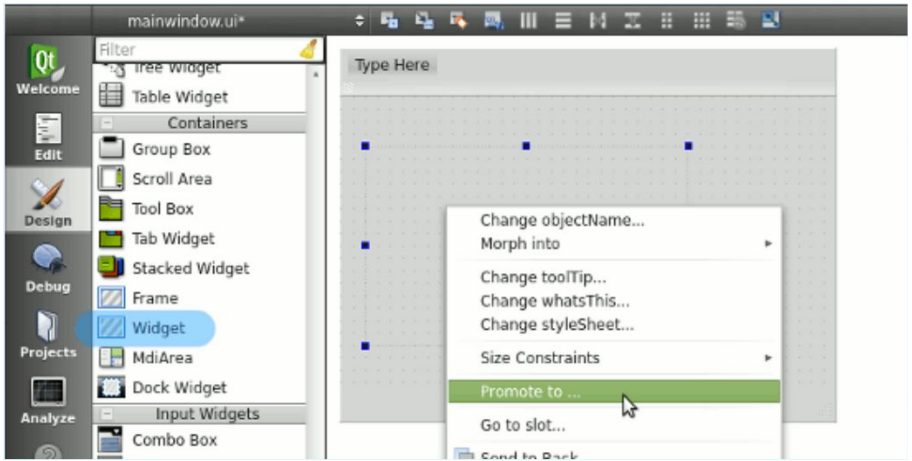
- MyLabel no conté codi C++, cal preprocessar-lo (MOC)
- Només cal afegir el .h i el .cpp al .pro

```
HEADERS+=MyLabel.h
SOURCES+=MyLabel.cpp
```

- Crida al MOC automàtica
- Es generen fitxers moc_*

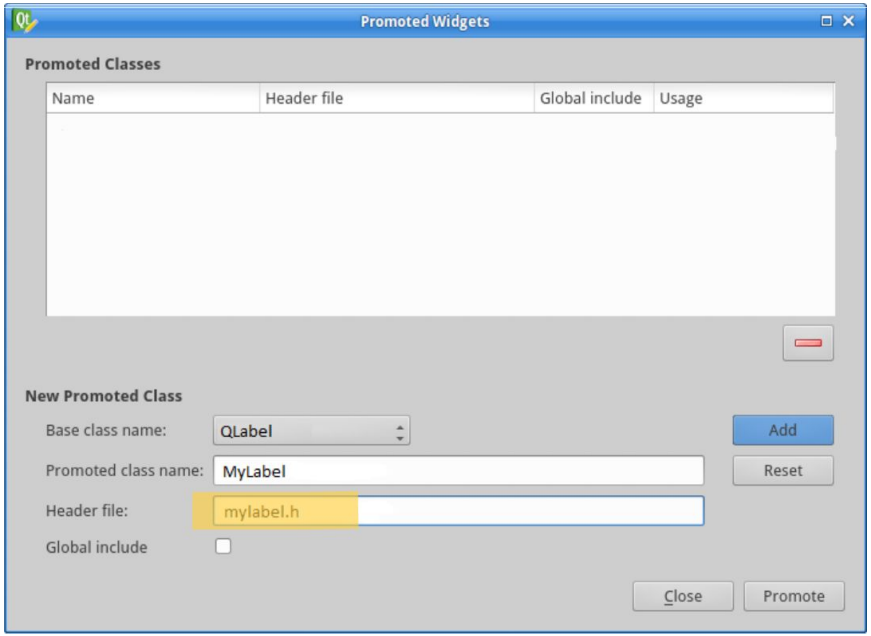
Llibreria Qt: Promote

- El resultat és un .h i un .cpp
- El designer no detecta la nova classe que hem creat
- Al designer, posar el widget de la classe base: QPushButton, QLCDNumber...
- Sobre el widget prémer el botó dret del ratolí i fer Promote to ...



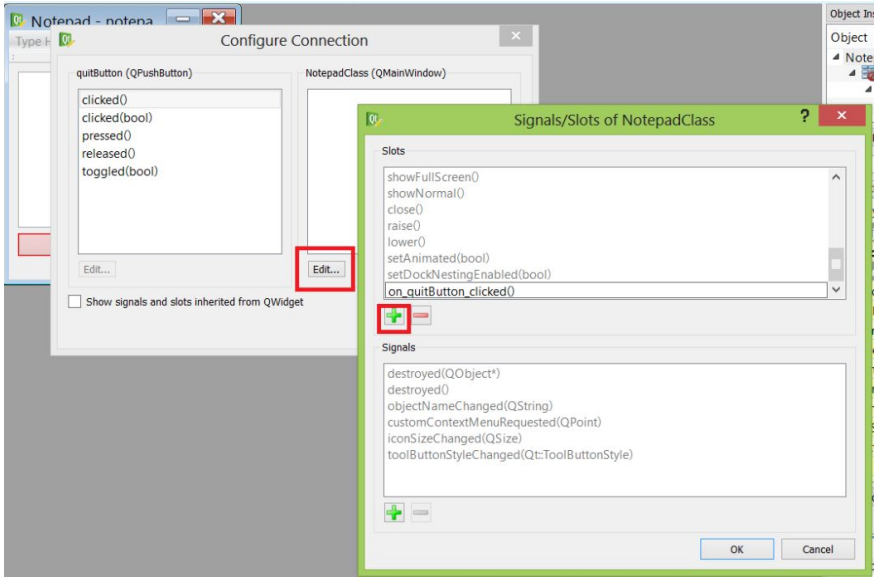
Llibreria Qt: Promote

- Posar la classe del nostre custom widget. Vigilar el nom del fitxer .h

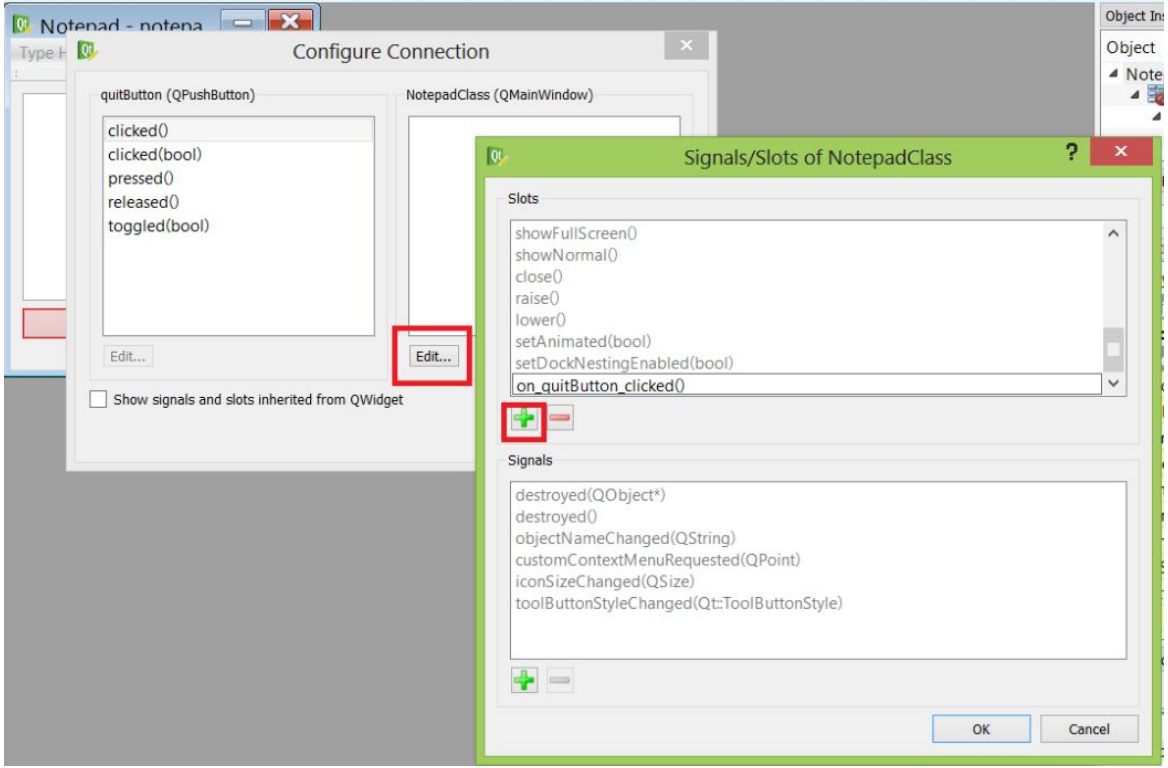


Llibreria Qt: Promote

- Cal connectar els signals/slots de la nosta classe amb altres signals/slots.
- El designer NO DETECTA els nous signals i slots del .h, cal afegir-los a mà al diàleg de connexió. VIGILAR ELS NOMS I ELS PARÀMETRES !!!!

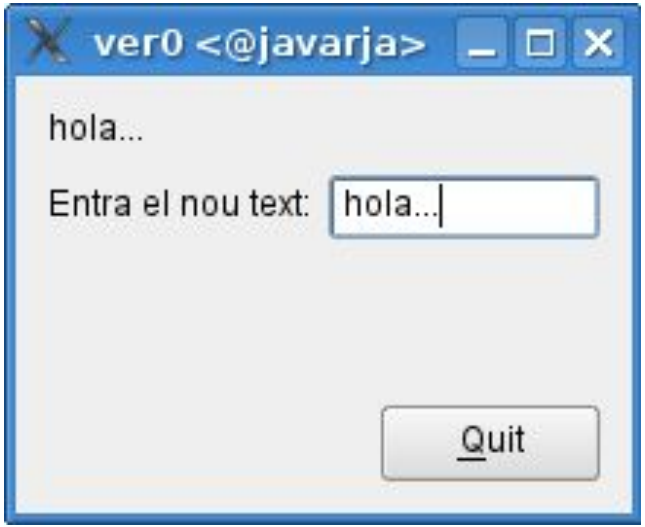


Llibreria Qt: Promote



Llibreria Qt: Problema 2

- Si volem que només copiï el text a l'etiqueta quan es fa <*return*>…



Signals QLineEdit:

- returnPressed ()

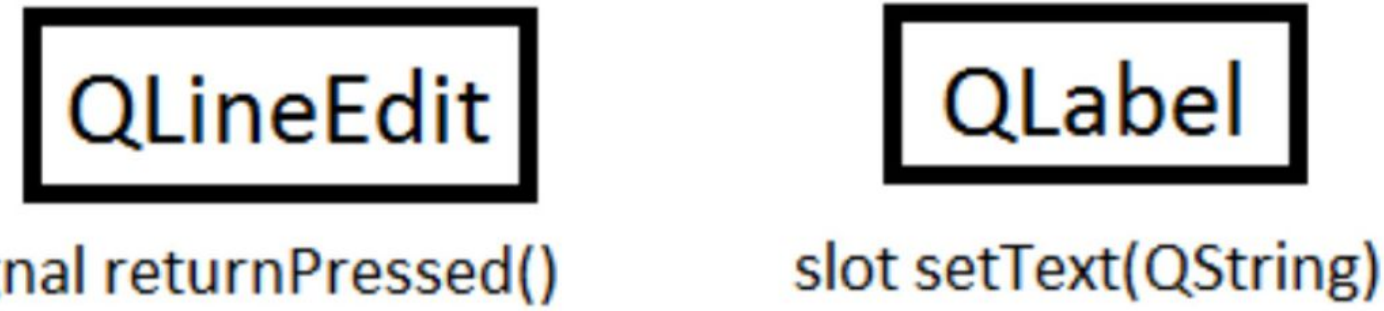
Slots QLabel:

- setText (QString)

NO ES PODEN CONNECTAR !

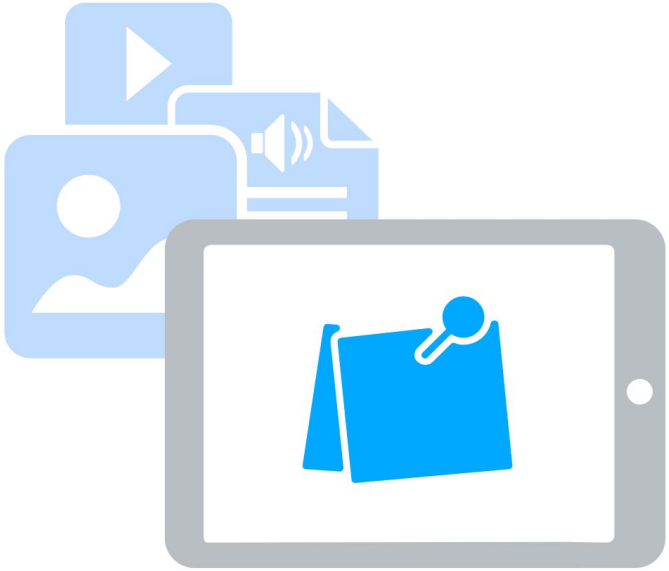
Llibreria Qt: Problema 2

- Com ho podem fer ?



No es poden connectar

Collaborate!



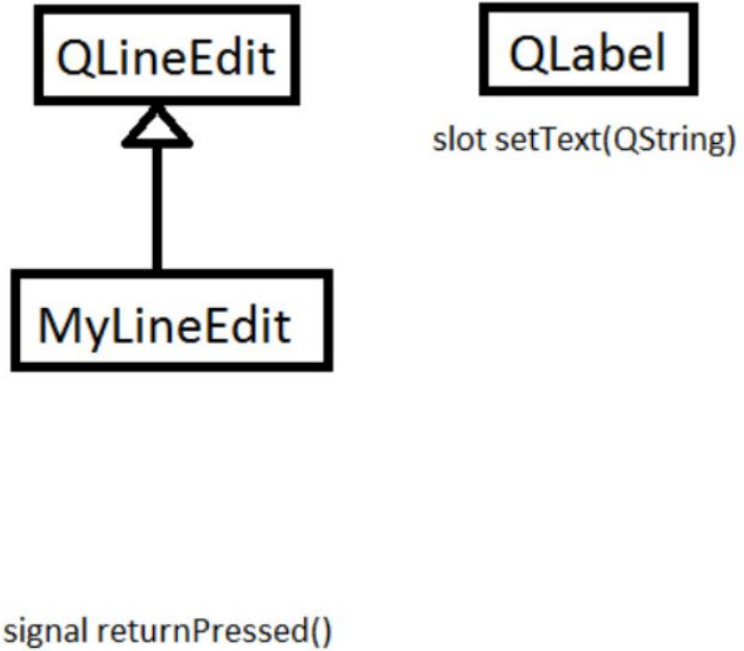
How to Edit

Click Edit This Slide in the plugin to make changes.

Don't have the Nearpod add-on? Open the "Add-ons" menu in Google Slides to install.

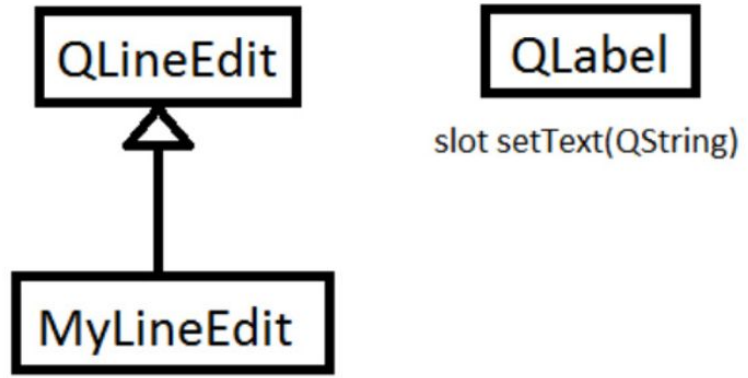
Llibreria Qt: Problema 2

- Com ho podem fer ? HEREDANT !



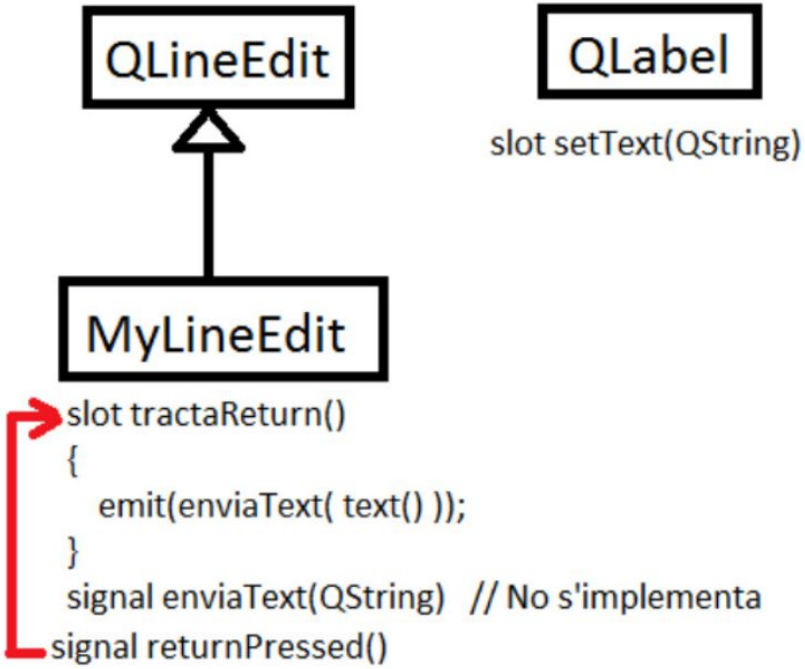
Llibreria Qt: Problema 2

- Com ho podem fer ? HEREDANT !



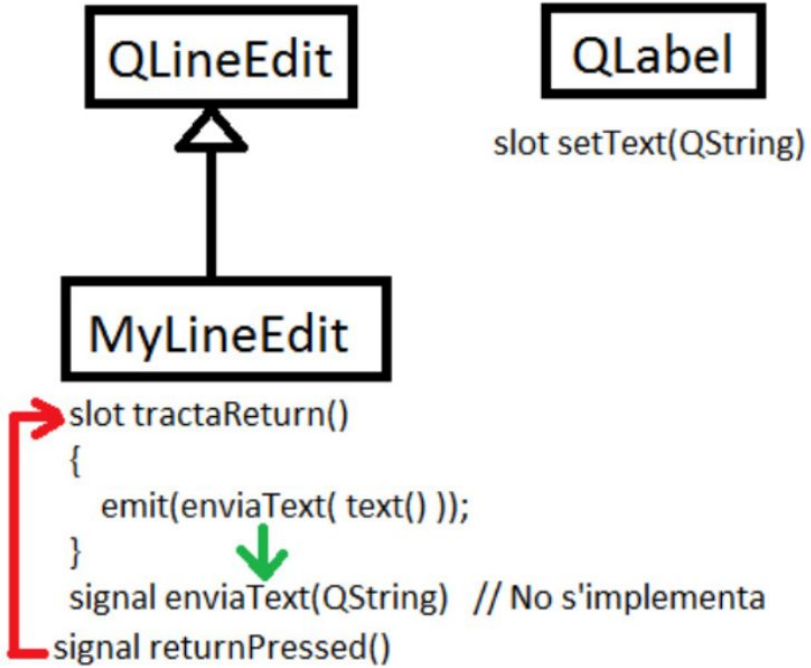
Llibreria Qt: Problema 2

- Com ho podem fer ? HEREDANT !



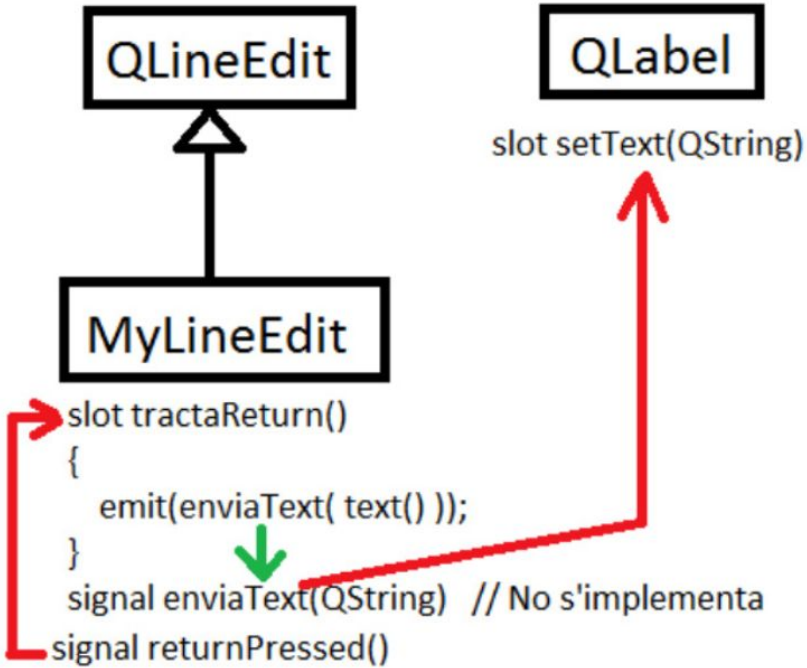
Llibreria Qt: Problema 2

- Com ho podem fer ? HEREDANT !



Llibreria Qt: Problema 2

- Com ho podem fer ? HEREDANT !



Llibreria Qt: Solució

- Heredar de la classe QLineEdit: MyLineEdit
- Crear un slot a MyLineEdit sense paràmetres: tractaReturn()
- Connectar returnPressed() amb tractaReturn()
- tractaReturn emet un signal que envia el text guardat al QLineEdit i que es pot consultar amb el mètode text(). Aquest mètode és heredat per MyLineEdit
- El signal s'ha de crear a MyLineEdit i ha d'enviar un QString: enviaText(QString)
- El signal es connecta a l'slot setText del Label

Llibreria Qt: Notes Importants

- Els slots els implementarem a ”MyLineEdit.cpp”
- Els signals no s’implementen però es poden llençar en qualsevol punt del codi cridant a la funció:

emit nom_signal(paràmetres)

- Els signals NOMÉS ES PODEN CONNECTAR a slots, no s'implementen.
- Quan s'emet el signal, s'executen tots els slots que hi estan connectats

emit enviarEnter(5)

Problema 2: MyLineEdit.h

```
#include <QLineEdit>

class MyLineEdit: public QLineEdit
{
      Q_OBJECT     //IMPORTANT
 public:
      MyLineEdit (QWidget *parent);
 public slots:     // IMPORTANT
      void tractaReturn ();
 signals:          // IMPORTANT
      void enviaText (const QString &);
};
```

Problema 2: MyLineEdit.cpp

```
#include "MyLineEdit.h"

// constructor

MyLineEdit::MyLineEdit(QWidget *parent): QLineEdit(parent) {

}

// implementació slots

void MyLineEdit::tractaReturn() {

    // Implementació de tractaReturn

    emit enviaText (text());

}
```

Llibreria Qt: La classe MyGLWidget

- Com podeu veure, la nostra classe d'OpenGL MyGLWidget, en realitat és una classe pròpia derivada de QOpenGLWidget de Qt…
  - Podeu veure que el .h inclou la macro Q_OBJECT
  - I que tenim el fitxer .h en el tag HEADERS del .pro
- Per tant podem usar-la per a afegir comportament si volem que es pugui lligar amb altres components de Qt (és a dir, podem afegir-li signals i slots)
- Recordeu afegir el "`makeCurrent()`" al principi de qualsevol slot que hagi d'usar codi OpenGL.

Llibreria Qt: Notes importants

- Els custom widgets no funcionen amb CTRL+R, cal compilar i executar des de la consola.
- Els LCDNumber han de tenir una mida una mica gran per a què es vegin en color.
- Feu servir stylesheet pels colors, millor que la classe QPalette.
- Si feu servir QtCreator vigilar amb el Shadow Build. Assegureu-vos que podeu compilar i executar després des de consola.
- Per a fer una neteja

  make distclean

- Usar l'assistant.

Feina per avui

- Programar el custom widget per les majúscules
- Programar el custom widget pel return
- Fer els exercicis del racó
- Modificar la interfície de l'aplicació MyGLWidget
  - RadioButton per canviar de càmer
  - 3 LCD per canviar el color de fons
  - Un slider per fer zoom