|  |  |
| --- | --- |
| minilogo | Introduction à la programmation QT : Fenêtre graphique |
| SPE - ISN | **Programmer une fenêtre basique avec QT Creator**. |
| **Durée du TP :** 1h30 | |
| **Organisation :** Les manipulations se feront à l’aide de QT Creator installé sur les ordinateurs | |

# Exécuter le logiciel QTCreator.

**Ce premier programme aura pour seuls objectifs d’apprendre à créer un interface graphique possédant un menu, un bouton et un champ de saisie**

On pourra se servir des nombreuses informations dans l’espace WEB consacré à QTCreator :

<http://www.siloged.fr/cours/QTCreator>

**Création des fichiers**

**1/** Création du projet et des fichier sources

* Cliquer sur ***Fichier/Nouveau Fichier ou projet***
* Choisir ***Application QT4 avec Widgets*** puis le bouton ***choose***
* Donner le nom ***TP1\_fenetre*** et sauver le fichier dans le dossier ***d:\ISN\TP1\_fenetre***
* Valider deux fois par ***suivant***
* Sur l’écran ***Information de la classe*** figure le nom que portera la fenêtre. Laisser la valeur par défaut, ne changez rien, cliquer sur le bouton***Suivant***
* Finir en cliquant sur le bouton ***Terminer***

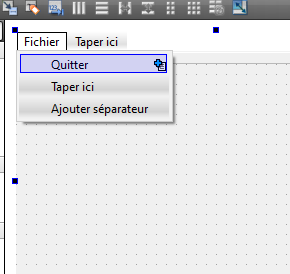
QTCreator a créé un ensemble de fichiers.   
***Nommer*** ces fichiers et ***donner*** leur rôle ?

|  |
| --- |
|  |

**Création du menu**

**2/** Double-cliquer sur le formulaire *mainwindow.ui*. ***Expliquer*** quelle est la réaction de QTCreator ?

|  |
| --- |
|  |

****

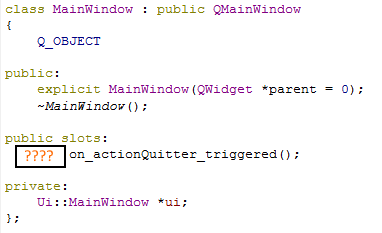
**3/** Dans la partie centrale de l’éditeur de formulaire (design) cliquer dans la barre de menu pour créer un menu principal Fichier et une commande Quitter. La commande Quitter est associée à un « type »(ce terme est impropre. On le corrigera ultérieurement).

***Donner*** le nom de la variable associée à cette commande ?

|  |
| --- |
|  |

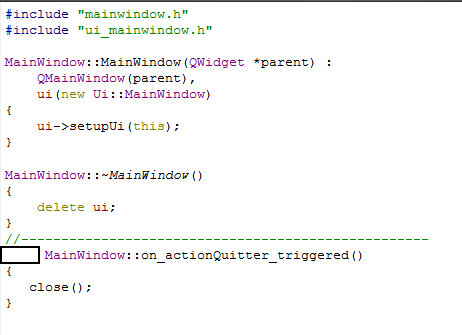
4/ ***Sauver*** tout (Ctrl+Maj+S) et ***ouvrir*** dans l’éditeur le fichier mainwindow.h. Exécuter le programme pour vérifier que la fenêtre est bien conforme à ce qu’elle doit être. La commande Quitter ne fonctionne pas car aucun code ne lui est associé

**Gérer l’évènement « clic »sur le menu**

5/ La procédure ***on\_actionQuitter\_triggered()*** va contenir le code permettant de quitter le programme.

***Modifier*** le code de mainwindow.h de la manière suivante.

***Remplacer*** les points d’interrogation par le mot clé adapté.

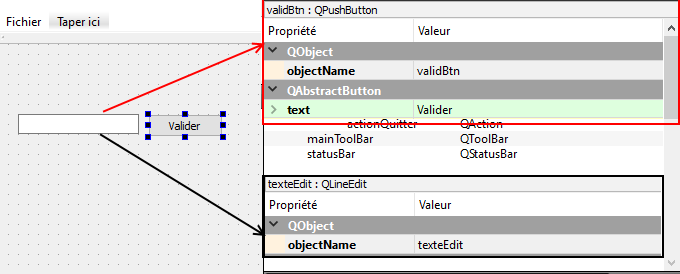
6/ Dans ***mainwindow.cpp*** créer la procédure déclarée précédemment.

***Tester*** le programme.

***Expliquer*** pourquoi….

|  |
| --- |
| il fallait déclarer la procédure avant de l’utiliser :  pourquoi la mettre dans les fichiers mainwindow.h et mainwindow.cpp |

**Ajout d’un bouton et d’un champ de saisie**

**7/** Retourner dans l’éditeur de formulaire et glisser un ***PushButton*** de la barre d’outil de gauche vers la fenêtre centrale. Faire de même pour le champ de saisie ***LineEdit***.

**8/** Changer le nom (objectName) du champ d’édition en ***texteEdit***

Changer le nom du bouton en ***validBtn***

Changer le texte (***text***) du bouton en Valider

**9/** Sauver et compiler pour voir le résultat.

**Gérer l’évènement « clic »sur le menu**

**10/** La méthode est presque la même.

Déclarer une procédure appelée **on\_validBtn\_clicked()** dans l’entête de mainwindow

**ATTENTION : l’évènement s’appelle clicked et pas triggered**

**11/** ***Modifier*** le code de mainwindow.cpp. ***Commenter*** chaque ligne en précisant sa fonction.

|  |  |
| --- | --- |
| #include "mainwindow.h"  #include "ui\_mainwindow.h"  #include <QDebug>  MainWindow::MainWindow(QWidget \*parent) :  QMainWindow(parent),  ui(new Ui::MainWindow)  {  ui->setupUi(this);  }  MainWindow::~*MainWindow*()  {  delete ui;  }  //---------------------------------------------------  void MainWindow::on\_actionQuitter\_triggered()  {  close();  }  //----------------------------------------------------  void MainWindow::on\_validBtn\_clicked()  {  qDebug()<<ui->texteEdit->text();  } |  |