

GPA789

# Analyse et conception orientées objet

Laboratoire 2 (version préliminaire a)

# 1 TABLE DES MATIÈRES

---

1	Introduction .....	1
2	Objectifs pédagogiques.....	1
3	Contenu du laboratoire.....	1
3.1	Spécifications des fonctionnalités et de l'interface utilisateur graphique .....	1
3.2	Présentation des contrôles graphiques .....	4
4	Annexe – Mise en place de la solution dans Visual Studio .....	8
5	Annexe – Informations techniques complémentaires.....	12
6	Annexe – Conception des interfaces utilisateurs graphiques.....	13
6.1	Explorateur de fichiers et consultation.....	13
6.2	Outil de génération de fichiers de sortie .....	13
6.3	Suppléments .....	14
6.4	Boîte de dialogue à propos .....	14
7	Annexe – Contrôles graphiques des IUG (widgets).....	15
7.1	Explorateur de fichiers et consultation.....	16
7.2	QDirectorySelector.....	16
7.3	QCppCommentViewer .....	16
7.4	Outil de génération de fichiers de sortie .....	17
7.5	QSelectedFileList.....	17
7.6	Suppléments .....	18
7.7	Boîte de dialogue à propos .....	18
8	Annexe – Interfaces utilisateurs graphiques à réaliser.....	19
8.1	Explorateur de fichiers et consultation.....	19
8.2	Outil de génération de fichiers de sortie .....	20
8.3	Suppléments et boîte de dialogue à propos .....	21

# 1 INTRODUCTION

---

Le programme XtractC développé dans le premier laboratoire de ce cours consistait à relever les commentaires contenus dans un fichier source du langage C ou C++. Ce programme prenait en paramètre, par la ligne de commande, un nom de fichier d'entrée et un nom de fichier de sortie. Le fichier d'entrée devant être un fichier source en langage C ou C++. XtractC fonctionne en mode console et ne peut traiter qu'un seul fichier source à la fois. Son utilisation est par conséquent difficile et peu efficace.

Dans ce deuxième laboratoire, nous allons créer un programme basé sur une interface utilisateur graphique (IUG) sous Windows. La création de cette interface sera réalisée à l'aide du cadre de travail (Framework) Qt. Par conséquent, nous utiliserons l'approche orientée objet de la bibliothèque Qt qui réalise l'encapsulation des éléments de l'IUG à l'aide de classes. Ainsi, tous les composants graphiques (communément appelé widgets) tels que les boutons, les fenêtres et les menus sont représentés par des classes appropriées. Le traitement des événements et des messages normalement associés à la programmation Windows est remplacé par le mécanisme de signaux et de slots.

L'IUG de XtractC sera appelée XtractCgui (XtractC Graphical User Interface) et on y retrouve trois tâches principales :

1. navigation et sélection de fichiers;
2. visualisation d'un fichier source ainsi que des commentaires associés;
3. traitement d'un ensemble de fichiers afin de produire plusieurs fichiers de sortie contenant les commentaires.

## 2 OBJECTIFS PÉDAGOGIQUES

---

Les objectifs pédagogiques de ce laboratoire sont énumérés ci-dessous en ordre décroissant d'importance:

- Apprendre la structure fondamentale d'une application Windows utilisant Qt.
- Comprendre le principe de programmation à l'aide d'un cadre de travail orienté-objet (Qt).
- Acquérir une bonne expérience dans l'utilisation des bibliothèques de classes.
- Connaître les éléments communs des interfaces graphiques (widget).
- Se familiariser avec l'environnement de développement Visual C++ et mieux comprendre comment mettre en place une solution incluant plusieurs projets.

## 3 CONTENU DU LABORATOIRE

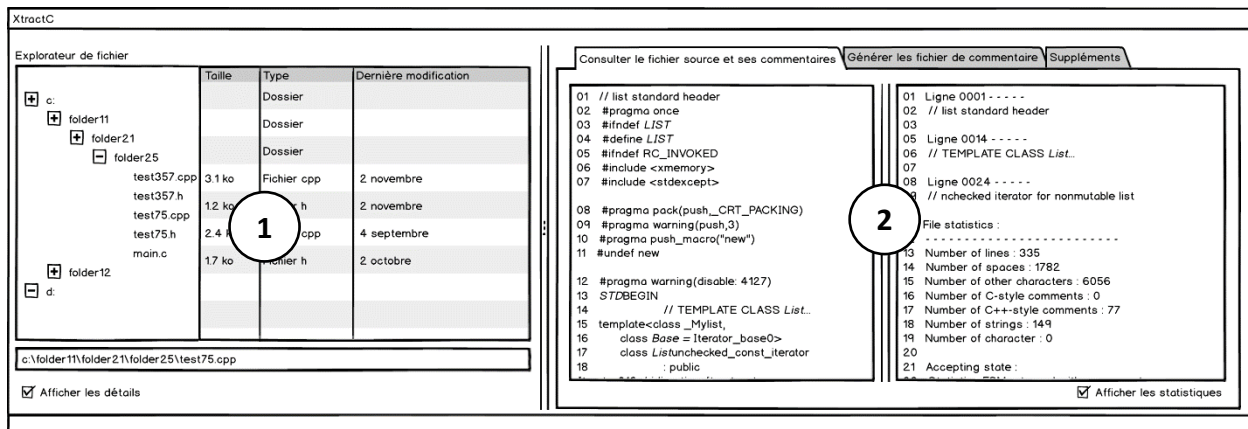
---

Cette section présente les explications détaillées de ce laboratoire. Lisez attentivement les exposés ci-dessous puisqu'ils vous aideront certainement dans la réalisation du projet de programmation.

### 3.1 SPÉCIFICATIONS DES FONCTIONNALITÉS ET DE L'INTERFACE UTILISATEUR GRAPHIQUE

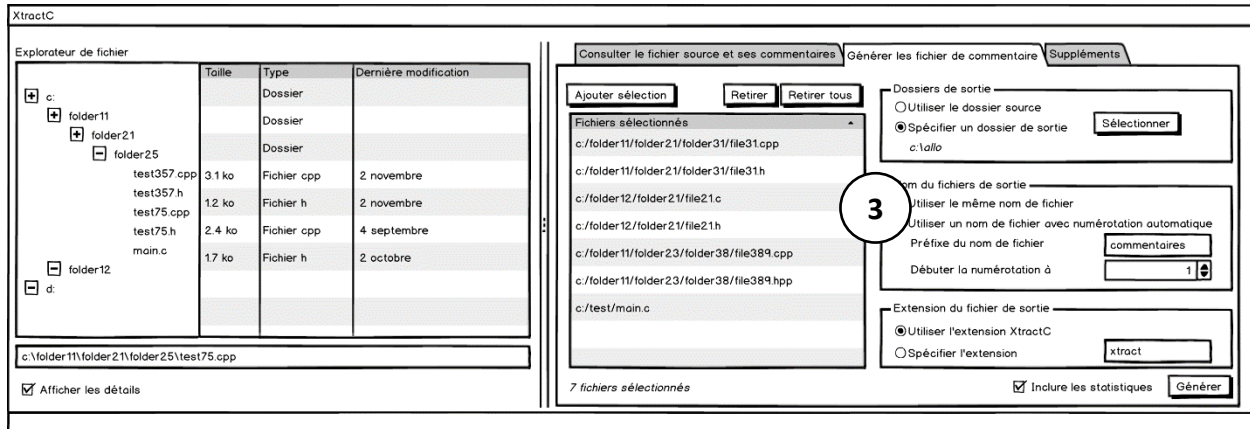
Les caractéristiques de cette interface graphique sont :

- Visualisation de contenu des disques et fichiers de l'ordinateur.
  - Les disques réseaux et leurs contenus doivent également être visibles par l'utilisateur.
  - La présentation graphique de cette visualisation doit utiliser une présentation arborescente en arbre semblable à celle de l'explorateur Windows et de logiciels équivalents.
  - Tous les fichiers sont affichés et peuvent être sélectionnés. Par contre, seul les fichiers standards du langage C++ sont autorisés à pouvoir être traités par les opérations subséquentes (\*.c, \*.h, \*.cpp, \*.hpp).
- Suivant la sélection d'un fichier, on doit pouvoir consulter le contenu du fichier et de ses commentaires.
  - Deux éditeurs textes doivent être disposés latéralement côte à côte avec l'affichage du fichier source à gauche et des commentaires à droite.
  - Ces éditeurs ne doivent pas permettre la modification du contenu mais plutôt offrir une consultation en lecteur seule.
  - L'affichage du code doit pouvoir offrir une représentation visuelle similaire à un éditeur C++ en indiquant les numéros de lignes ainsi qu'en rehaussant la syntaxe des éléments propres au langage.
  - Pour la consultation des commentaires, il doit être possible d'afficher ou non les statistiques du fichier source.

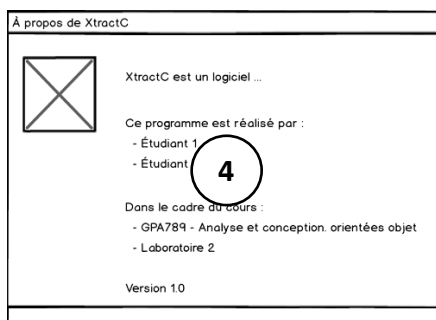
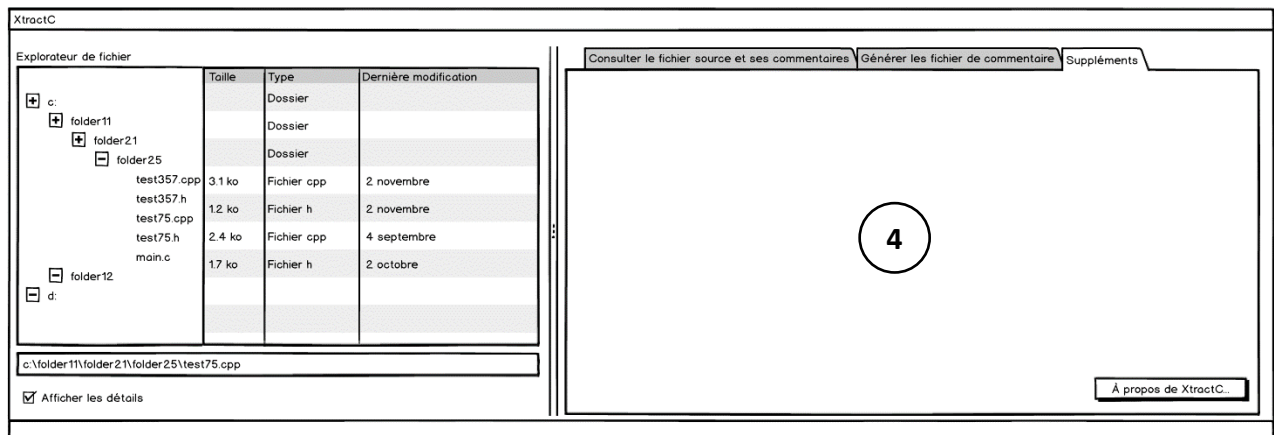


- Sélection d'un ou plusieurs fichiers afin de produire des fichiers de sortie.
  - Gestion des fichiers à traiter en offrant la possibilité de :
    - Ajouter les fichiers sélectionnés dans l'arborescence principale (partie 1). Seul les fichiers du langage peuvent être insérés (\*.c, \*.h, \*.cpp, \*.hpp). De plus, les doublons sont interdits dans la liste.
    - Retirer les éléments sélectionnés de la liste des fichiers retenus (partie 3).
    - Retirer tous les éléments sélectionnés.
  - Définition du dossier de réception des fichiers de sortie. L'utilisateur doit choisir entre le dossier source ou un autre dossier.
  - Définition du nom des fichiers de sortie. L'utilisateur doit choisir entre le nom du fichier source ou un nom généré par un préfixe et un numéro généré automatiquement selon la liste de fichiers retenus.

- Définition de l'extension du fichier de sortie. L'utilisateur doit choisir entre le choix par défaut (extension XtractC) ou un de son choix.
- Pour la génération des fichiers de sortie, il doit être possible de produire ou non les statistiques du fichier source.



- Présentation d'un panneau de message indiquant les informations suivantes sur le programme :
  - nom du programme;
  - brève description;
  - noms des étudiants;
  - indication du contexte (cours et laboratoire);
  - version du logiciel.



## 3.2 TECHNIQUES DE PROGRAMMATION ET PRATIQUES RECOMMANDÉES AVEC QT

Comme pour plusieurs bibliothèques similaires qui sont basées sur la programmation orientée objet, Qt exploite le plein potentiel de cette technique de développement.

Lors du développement d'un projet, il est très fréquent d'avoir à ajouter des fonctionnalités supplémentaires aux classes existantes de Qt. Par exemple, on peut vouloir ajouter un cadre de couleur à une zone de texte pour représenter un état spécifique.

L'une des stratégies les plus importantes est la nature extensible des classes. Il est donc possible et recommandée de créer de nouvelles classes héritant d'autre classe afin de redéfinir certains comportements et de tirer avantage de toute l'infrastructure mise en place. C'est la responsabilité du programmeur novice de développer le réflexe d'adopter cette technique de programmation efficace et modulaire.

Il est aussi fréquent de réunir plusieurs contrôles graphiques sous un seul afin de regrouper plusieurs widgets nécessaires à une opération spécifique. Ainsi, la stratégie consiste à créer un nouveau widget héritant de la classe QWidget. Ce nouveau widget pourrait posséder ses propres comportements mais aussi plusieurs autres widgets par composition. Ainsi, le widget parent devient responsable et encapsule toutes les opérations nécessaires à un comportement global plus cohérent selon le contexte.

Cette technique de programmation permet de créer de nouveau widget plus modulaire et facilite grandement la réutilisabilité mais aussi le développement en compartimentant les différents défis techniques.

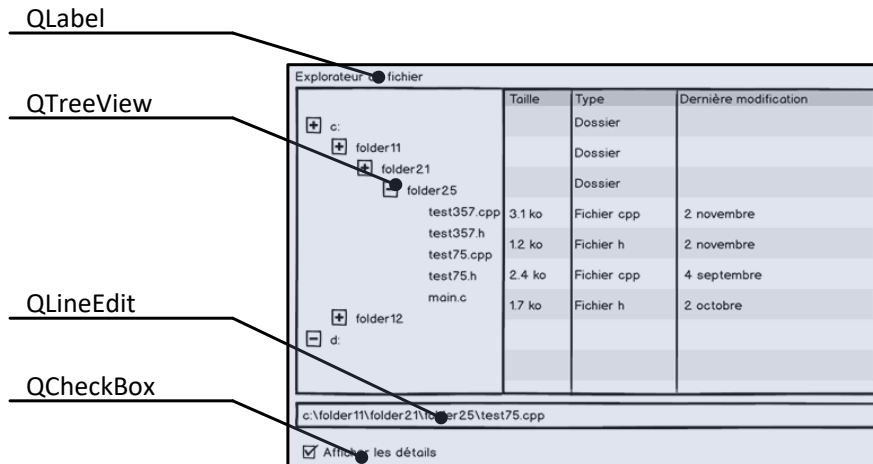
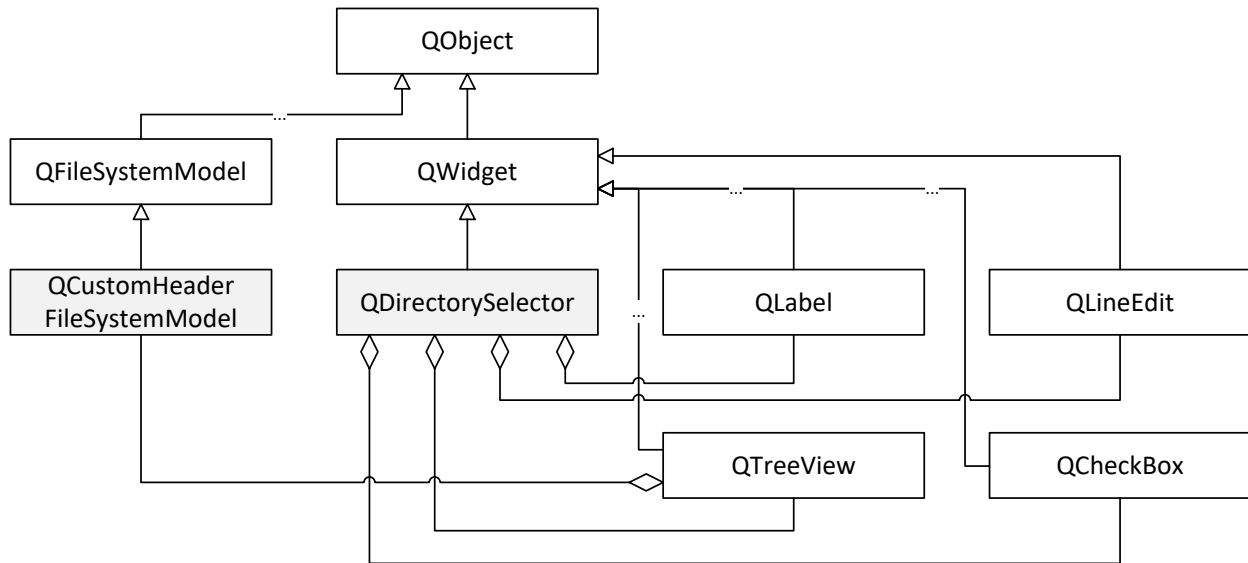
Pour ce laboratoire, on vous encourage à développer ces techniques en observant quelques classes données et dans le développement que vous devez faire.

## 3.3 PRÉSENTATION DES CONTRÔLES GRAPHIQUES

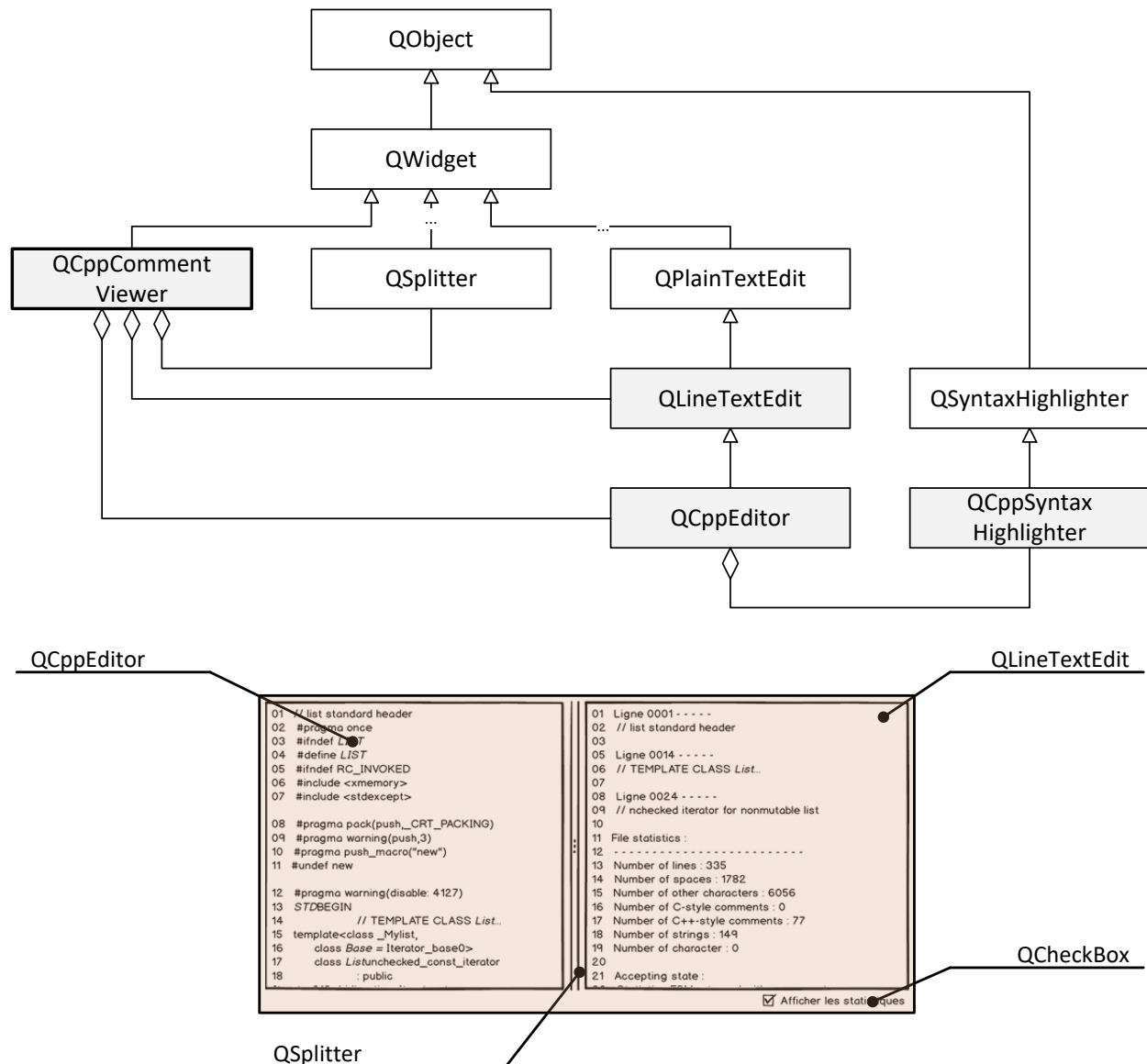
La section 8 intitulée Annexe – Contrôles graphiques des IUG (widgets), présente tous les widgets utilisés pour le développement de l'application. La présentation inclue une disposition détaillée de chaque contrôle.

On remarque que la conception de l'interface possède plusieurs widgets combinés qui vous sont donnés. Parmi les classes qui vous sont données, il en existe trois plus importantes :

- QDirectorySelector :
  - est un widget;
  - hérite de la classe QWidget
  - est un amalgame de quatre widgets : QLabel, QTreeView, QLineEdit et QCheckBox
  - a pour mandat
    - d'afficher le contenu des dossiers et fichiers sur les disques;
    - possède une zone texte interactive permettant de manipuler le dossier;
    - offre une case à cocher permettant de masquer ou non les informations détaillées.
  - Utilise une autre classe créée afin de permettre des noms de colonnes différents de la classe de base : QCustomHeaderFileSystemModel

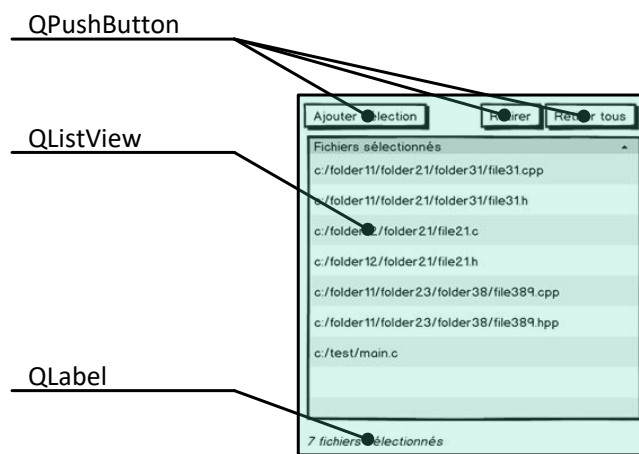
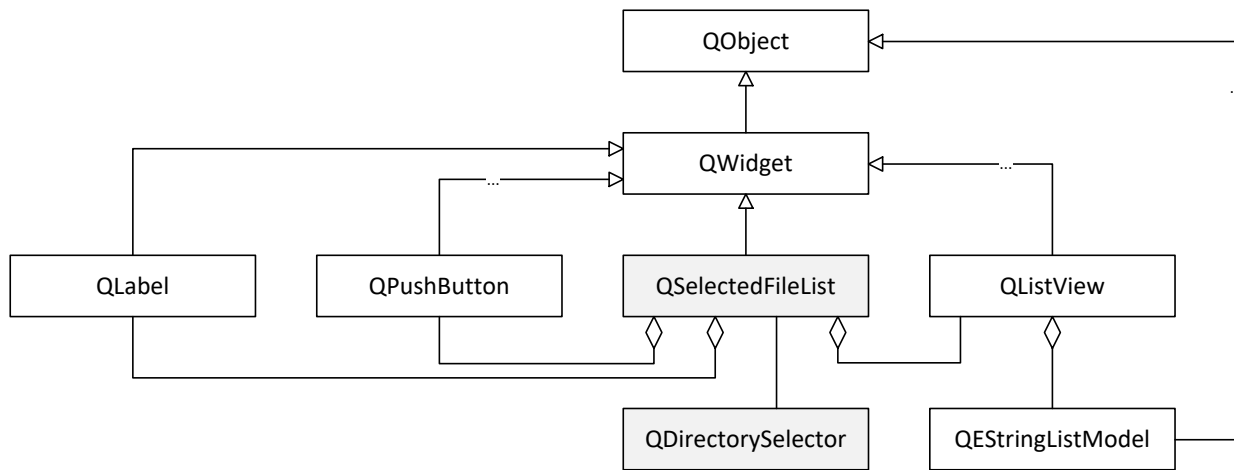


- QCppCommentViewer :
  - Est un widget.
  - Hérite de QWidget.
  - Est un amalgame de quatre widgets : QCppEditor, QLineTextEdit, QSplitter et QCheckBox;
  - A pour mandat :
    - d'afficher le fichier et ses commentaires dans les deux contrôles prévus à cet effet
    - réagir au QDirectorySelector si ce dernier subit un click. Dans ce cas, on doit valider la conformité du fichier et le traiter en conséquence.
    - offre une case à cocher permettant d'inclure ou non les données statistiques dans l'éditeur prévu aux commentaires.
  - Utilise deux classes créées :
    - QLineTextEdit : un éditeur texte avec les numéros de ligne (hérite de QPlainTextEdit).
    - QCppEditor : un éditeur avec numéro de ligne et le rehaussement syntaxique du langage C++ (hérite de QLineTextEdit et utilise ).



- **QSelectedFileList :**
  - Est un widget.
  - Hérite de **QWidget**.
  - Est un amalgame de cinq widgets : **QTreeView**, 3 x **QPushButton** et **QLabel**;
  - A pour mandat :
    - D'afficher la liste des fichiers sélectionnés.
    - Ajouter 1 ou plusieurs items, supprimer 1 ou plusieurs items ou supprimer tout les items de la liste.
    - Valide les entrées pour qu'elles soient des fichiers valides.
    - S'assure qu'il n'y ait pas de doublon dans la liste.





## 4 TÂCHES À EFFECTUER

On suggère de réaliser ce travail par les 4 étapes suivantes :

1. Faire la mise en place de la solution (à la section 5, intitulé Annexe – Mise en place de la solution dans Visual Studio, vous trouverez toute les étapes liées à cette tâche).
2. Assurez-vous que vous êtes capable de compiler une application vide.
3. Réalisez l'implantation de la base de l'interface usager avec la partie exploration et consultation (voir la section 8.1).
4. Réalisez l'implantation du 2<sup>e</sup> onglet servant à la génération de fichier de sortie (voir la section 8.4).
5. Réalisez l'implantation du 3<sup>e</sup> onglet servant à afficher la boîte de dialogue à propos (voir les sections 8.6 et 8.7).
6. Finalisez les derniers détails.

## 5 ANNEXE – MISE EN PLACE DE LA SOLUTION DANS VISUAL STUDIO

---

À l'aide de Visual Studio, vous devez mettre en place une solution plus étayée considérant les pratiques de l'industrie dans le contexte de projet d'envergure. Évidemment, pour ce projet, l'approche proposée n'est pas essentielle mais elle reste pertinente et surtout très efficace pédagogiquement.

L'idée est de diviser tous ce qui a été donné au laboratoire 1 en trois sous projets et d'en ajouter un autre pour le laboratoire 2. Voici le découpage à réaliser :

- Solution XtractX
  - Projet Finite State Machine [bibliothèque statique] : possède toutes les classes associées aux machines à états finis. Cette approche permet de réutiliser simplement ces modules dans d'autres projets complètement différents.
  - Projet XtractC [bibliothèque statique] : possède toutes les classes associées à la construction d'une machine à états finis dédiée à l'extraction des commentaires en C++. Ces classes réalisent tout le travail d'extraction mais ne sont pas implémentées dans un programme en particulier.
  - Projet XtractCmd [programme exécutable – interface console] : implémente les classes de la bibliothèque statique XtractC dans un programme en mode console. C'est le programme donné au laboratoire 1.
  - Projet XtractGui [programme exécutable – interface graphique] : implémente les classes de la bibliothèque statique XtractC dans un programme en mode graphique avec plusieurs fonctionnalités pertinentes. C'est le programme à faire au laboratoire 2.

Les étapes pour mettre en place une telle solution sont nombreuses mais en valent la peine dans la réalisation d'un projet d'envergure. Voici les grandes lignes :

1. Créer une nouvelle solution. Puisqu'on va créer une solution avec un premier projet, il importe de créer le projet de cette partie. Le premier projet consiste à mettre en place une bibliothèque statique pour le module Finite State Machine. Voici les étapes :
  - a. File / New / Project
  - b. Win32 Console Application
  - c. Name : FiniteStateMachine
  - d. Solution name : XtractX
2. Définissez les paramètres du premier projet :
  - a. Application type : Static library
  - b. Additional options : retirer toutes les options
3. Mettez en place le premier projet :
  - a. Décompressez le fichier FiniteStateMachine.zip et copiez son contenu dans le dossier .../XtractX/FiniteStateMachine/
  - b. Ajoutez tous les filtres de ce projet dans l'explorateur de solution de Visual Studio :
    - FiniteStateMachine
      - ...
      - Build Events
      - Header Files
      - Match Symbol

- State
  - Transition
- Source Files
  - Match Symbol
  - State
  - Transition
- c. Importer tous les fichiers dans les filtres appropriés (identique au laboratoire 1). En plus, on ajoute les deux fichiers suivants dans le filtre Build Events (ils se trouvent dans le sous dossier tools) :
  - PostBuildEventDebug.bat
  - PostBuildEventRelease.bat
- d. Configurez les propriétés du projet :
  - Build Events :
    - Post-Build Event
      - tools/PostBuildEventDebug.bat (en mode debug/x64)
      - tools/PostBuildEventRelease.bat (en mode release/x64)
- 4. Mettez en place le deuxième projet XtractC :
  - a. Dans l'explorateur de solution, à l'aide du menu contextuel sur la solution, appuyez sur Add/New Project...
  - b. Donnez le nom de projet XtractC et choisissez Win32 Console Application.
  - c. Définissez les paramètres du projet :
    - Application type : Static library
    - Additional options : retirer toutes les options
  - d. Décompressez le fichier XtractC.zip et copiez sont contenu dans le dossier ../XtractX/XtractC /
  - e. Ajoutez le filtre Build Events.
  - f. Insérez tous les fichiers dans les filtres appropriés :
    - Build Events : tous les \*.bat
    - Header Files : tous les \*.h
    - Source Files : tous les \*.cpp
  - g. Configurez les propriétés du projet :
    - Build Events :
      - Post-Build Event
        - tools/PostBuildEventDebug.bat (en mode debug/x64)
        - tools/PostBuildEventRelease.bat (en mode release/x64)
    - C/C++ / General :
      - Additional Include Directory : ../FiniteStateMachine (pour les deux configuration Debug et Release)
    - Librarian / General :
      - Additional Dependencies :
        - FiniteStateMachine\_x64d.lib (pour la configuration Debug)
        - FiniteStateMachine\_x64.lib (pour la configuration Release)

- Additionnal Library Directory : ../lib (pour les deux configuration Debug et Release)
5. Mettez en place le troisième projet XtractCmd :
    - a. Dans l'explorateur de solution, à l'aide du menu contextuel sur la solution, appuyez sur Add/New Project...
    - b. Donnez le nom de projet XtractCmd et choisissez Win32 Console Application.
    - c. Définissez les paramètres du projet :
      - Application type : Win32 Console Application
      - Additionnal options : sélectionnez Empty projet et retirez toutes les autres options
    - d. Décompressez le fichier XtractCmd.zip et copiez son contenu dans le dossier ../XtractX/XtractCmd /
    - e. Ajoutez le filtre Build Events.
    - f. Insérez tous les fichiers dans les filtres appropriés :
      - Build Events : tous les \*.bat
      - Source Files : main.cpp
    - g. Configurez les propriétés du projet :
      - Build Events :
        - Post-Build Event
          - tools/PostBuildEventDebug.bat (en mode debug/x64)
          - tools/PostBuildEventRelease.bat (en mode release/x64)
      - Debugging :
        - Command Arguments : ../tests/Test\_CppComments.cpp
      - C/C++ / General :
        - Additionnal Include Directory : ../XtractC;../FiniteStateMachine (pour les deux configuration Debug et Release)
      - Linker / General :
        - Additionnal Library Directory : ../lib (pour les deux configuration Debug et Release)
      - Linker / Input :
        - Ajoutez XtractC\_x64d.lib (pour la configuration Debug)
        - Ajoutez XtractC\_x64.lib (pour la configuration Release)
  6. Mettez en place le quatrième projet XtractGui :
    - a. Dans l'explorateur de solution, à l'aide du menu contextuel sur la solution, appuyez sur Add/New Project...
    - b. Donnez le nom de projet XtractGui et choisissez Qt5 Projets / Qt Application.
    - c. Définissez les paramètres du projet : les paramètres par défaut sont satisfaisants
    - d. Décompressez le fichier XtractGui.zip et copiez son contenu dans le dossier ../XtractX/XtractGui /
    - e. Ajoutez le filtre Build Events.
    - f. Insérez tous les fichiers dans les filtres appropriés :
      - Build Events : tous les \*.bat
      - Header Files : tous les \*.h

- Source Files : tous les \*.cpp
- g. Configurez les propriétés du projet :
  - Build Events :
    - Post-Build Event
      - tools/PostBuildEventDebug.bat (en mode debug/x64)
      - tools/PostBuildEventRelease.bat (en mode release/x64)
  - C/C++ / General :
    - Additionnal Include Directory : ../XtractC;../FiniteStateMachine (pour les deux configuration Debug et Release)
  - Linker / General :
    - Additionnal Library Directory : ../lib (pour les deux configuration Debug et Release)
  - Linker / Input :
    - Ajoutez XtractC\_x64d.lib (pour la configuration Debug)
    - Ajoutez XtractC\_x64.lib (pour la configuration Release)
- 7. Finalisez la solution :
  - a. Créez les dossiers suivants dans la racine de la solution :
    - ../XtractX/bin
    - ../XtractX/lib
    - ../XtractX/tests
    - Décompressez le fichiers donné tests.zip et copiez sont contenu dans le dossier ../XtractX/tests
  - b. Déterminez l'ordre de dépendance des projets les uns par rapport aux autres :
    - Dans le menu Project / Project Dependencies..., assurez-vous d'avoir dans le deuxième onglet l'ordre suivant :
      - FiniteStateMachine
      - XtractC
      - XtractCcmd
      - XtractCgui

## 6 ANNEXE – INFORMATIONS TECHNIQUES COMPLÉMENTAIRES

---

Voici quelques suggestions concernant ce laboratoire :

1. La librairie Qt est impossible à résumer dans ce document. Il est fondamental que vous soyez capable de trouver l'information sur les différentes classe et stratégies de développement avec Qt. Heureusement pour vous, la documentation en ligne est particulièrement bien rédigée et la pratique rend la chose de plus en plus accessible.
2. La création d'une nouvelle application crée automatiquement une fenêtre principale avec trois contrôles graphiques : `QMenuBar` nommé `menuBar`, `QToolBar` nommé `mainToolBar` et `QStatusBar` nommé `statusBar`. Or, pour ce projet, le menu et la barre d'outils sont inutilisés. Il suffit d'ouvrir le fichier `XtractCgui.ui` avec le logiciel Qt Designer pour supprimer ces deux constituants.
3. En plus de tous les widgets dont vous devez gérer les signaux et slot, vous devez ajouter ces connections :
  - a. `QDirectorySelector::fileSelected` ▷ `QCppCommentViewer::setFile`
  - b. `QCppCommentViewer::eventSignaled` ▷ `QStatusBar::showMessage`
4. Lors de l'initialisation des classes `QCppCommentViewer` et `QSelectedFileList`, vous devez préciser les extensions de fichier acceptées avec ces fonctions de la façon suivante :
  - a. `QCppCommentViewer::setFileSuffixFilter({ "c", "h", "cpp", "hpp" });`
  - b. `QSelectedFileList::setFileSuffixFilter({ "c", "h", "cpp", "hpp" });`
5. N'oubliez pas d'ajouter l'icône fourni dans le fichier de ressource du projet Qt. Ensuite, vous pouvez attribuer cet icône par la ligne de code suivante directement dans le constructeur de votre classe `QMainAppilcation` : `setWindowIcon(QIcon(":/XtractCgui/icon"));`
6. Le rehausseur syntaxique inclue dans la classe `QCppEditor` est incomplet et fonctionne partiellement. Cet outil n'utilise pas d'interpréteur de code mais plutôt tente de faire correspondre des parties du document à des expressions régulières. Il est donc impossible de produire un rehausseur syntaxique sans erreur. Toutefois, le résultat obtenu est satisfaisant pour ce laboratoire.
7. Pour faire afficher la boîte de dialogue À propos de XtractC, utiliser la fonction statique suivante : `QMessageBox::about(...)`

## 7 ANNEXE – CONCEPTION DES INTERFACES UTILISATEURS GRAPHIQUES

### 7.1 EXPLORATEUR DE FICHIERS ET CONSULTATION

XtractC

Explorateur de fichier

	Taille	Type	Dernière modification
c:		Dossier	
+ folder11		Dossier	
+ folder21		Dossier	
- folder25		Dossier	
test357.cpp	3.1 ko	Fichier cpp	2 novembre
test357.h	1.2 ko	Fichier h	2 novembre
test75.cpp	2.4 ko	Fichier cpp	4 septembre
test75.h	1.7 ko	Fichier h	2 octobre
main.c			
- folder12			
d:			

c:\folder11\folder21\folder25\test75.cpp

☒ Afficher les détails

Consulter le fichier source et ses commentaires

```
01 // list standard header
02 #pragma once
03 #ifndef LIST
04 #define LIST
05 #ifndef RC_INVOKED
06 #include <xmemory>
07 #include <stdexcept>
08 #pragma pack(push,_CRT_PACKING)
09 #pragma warning(push,3)
10 #pragma push_macro("new")
11 #undef new
12 #pragma warning(disable: 4127)
13 #ifndef BEGIN
14 // TEMPLATE CLASS List..
15 template<class _Mylist,
16         class Base = Iterator_base0>
17     class Listunchecked_const_iterator
18         : public
19         {
20     public:
21     // ...
22 }
```

Générer les fichier de commentaire

Suppléments

01 Ligne 0001 - - - -  
02 // list standard header  
03  
04 Ligne 0014 - - - -  
05 // TEMPLATE CLASS List..  
06  
07  
08 Ligne 0024 - - - -  
09 // nchecked iterator for nonmutable list  
10  
11 File statistics :  
12 -----  
13 Number of lines : 335  
14 Number of spaces : 1782  
15 Number of other characters : 6056  
16 Number of C-style comments : 0  
17 Number of C++-style comments : 77  
18 Number of strings : 149  
19 Number of character : 0  
20  
21 Accepting state :  
22

☒ Afficher les statistiques

### 7.2 OUTIL DE GÉNÉRATION DE FICHIERS DE SORTIE

XtractC

Explorateur de fichier

	Taille	Type	Dernière modification
c:		Dossier	
+ folder11		Dossier	
+ folder21		Dossier	
- folder25		Dossier	
test357.cpp	3.1 ko	Fichier cpp	2 novembre
test357.h	1.2 ko	Fichier h	2 novembre
test75.cpp	2.4 ko	Fichier cpp	4 septembre
test75.h	1.7 ko	Fichier h	2 octobre
main.c			
- folder12			
d:			

c:\folder11\folder21\folder25\test75.cpp

☒ Afficher les détails

Consulter le fichier source et ses commentaires

Ajouter sélection

Retirer

Retirer tous

Fichiers sélectionnés

- c:/folder11/folder21/folder31/file31.cpp
- c:/folder11/folder21/folder31/file31.h
- c:/folder12/folder21/file21.c
- c:/folder12/folder21/file21.h
- c:/folder11/folder23/folder38/file389.cpp
- c:/folder11/folder23/folder38/file389.hpp
- c:/test/main.c

7 fichiers sélectionnés

Générer les fichier de commentaire

Suppléments

Dossiers de sortie

☐ Utiliser le dossier source

☒ Spécifier un dossier de sortie

Sélectionner

c:\allo

Nom du fichiers de sortie

☐ Utiliser le même nom de fichier

☒ Utiliser un nom de fichier avec numérotation automatique

Préfixe du nom de fichier

commentaires

Débuter la numérotation à

1

Extension du fichier de sortie

☒ Utiliser l'extension XtractC

☐ Spécifier l'extension

xtract

☒ Inclure les statistiques

Générer

## 7.3 SUPPLÉMENTS

XtractC

Explorateur de fichier

+

c:

+

folder11

+

folder21

-

folder25test357.cpptest357.h  
test75.cpptest75.h  
main.c

-

folder12

-

d:

Taille

Type

Dernière modification

3.1 ko

Fichier cpp

2 novembre

1.2 ko

Fichier h

2 novembre

2.4 ko

Fichier cpp

4 septembre

1.7 ko

Fichier h

2 octobre

c:\folder11\folder21\folder25\test75.cpp

☒ Afficher les détails

Consulter le fichier source et ses commentaires


Générer les fichier de commentaire

Suppléments

À propos de XtractC...

## 7.4 BOÎTE DE DIALOGUE À PROPOS

À propos de XtractC



XtractC est un logiciel ...

Ce programme est réalisé par :

- Étudiant 1
- Étudiant 2

Dans le cadre du cours :

- GPA789 - Analyse et conception, orientées objet
- Laboratoire 2

Version 1.0



## 8 ANNEXE – CONTRÔLES GRAPHIQUES DES IUG (WIDGETS)

---

Cette section présente les différents widgets à utiliser pour réaliser l'application. Les contrôles graphiques proposés sont les widgets standards de la librairie Qt. Toutefois, trois d'entre eux sont des widgets personnalisés qui vous sont donnés. Dans les illustrations suivantes, ces classes sont indiquées par les couleurs bleu, orange et vert.

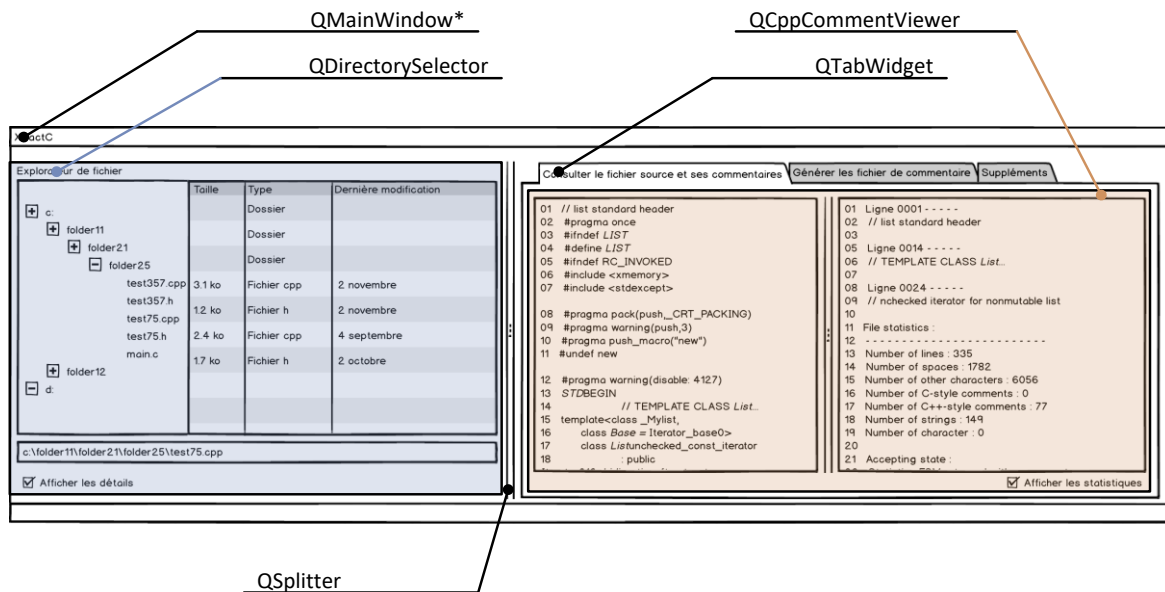
Les widgets standards utilisés :

- QMainWindow\* (indirectement par héritage de la classe principale du programme)
- QTabWidget
- QSplitter
- QLabel
- QTreeView
- QLineEdit
- QSpinBox
- QCheckBox
- QRadioButton
- QPushButton
- QGroupBox
- QMessageBox

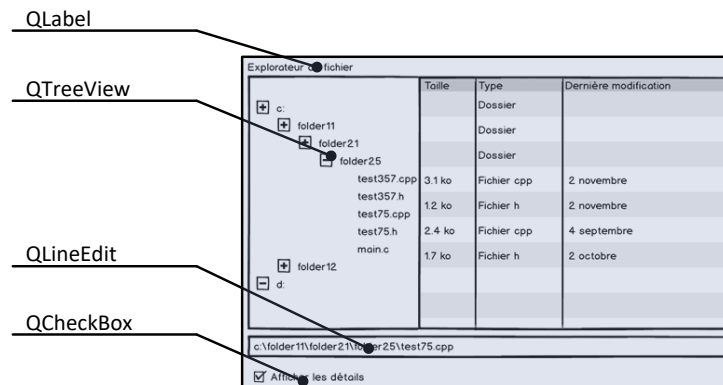
Les widgets personnalisés sont :

- QDirectorySelector (illustré en bleu)
- QCppCommentViewer (illustré en orange qui inclus QLineTextEdit et QCppEditor)
- QSelectedFileList (illustré en vert)

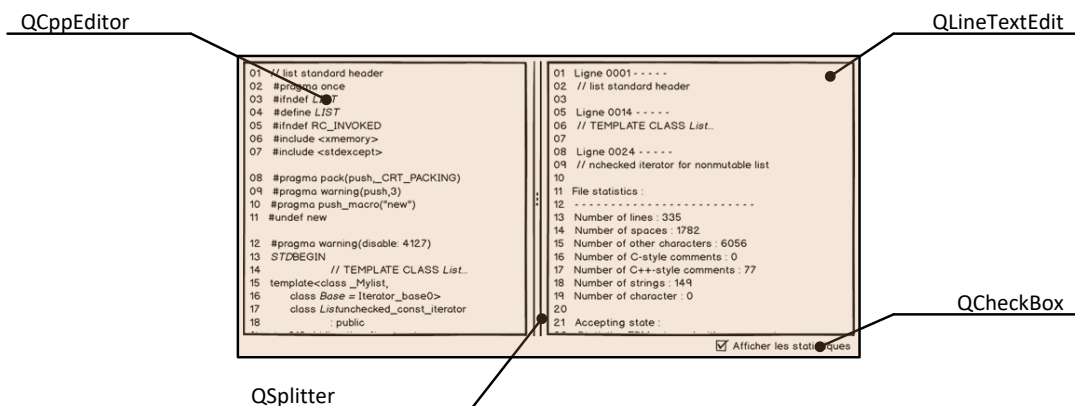
## 8.1 EXPLORATEUR DE FICHIERS ET CONSULTATION



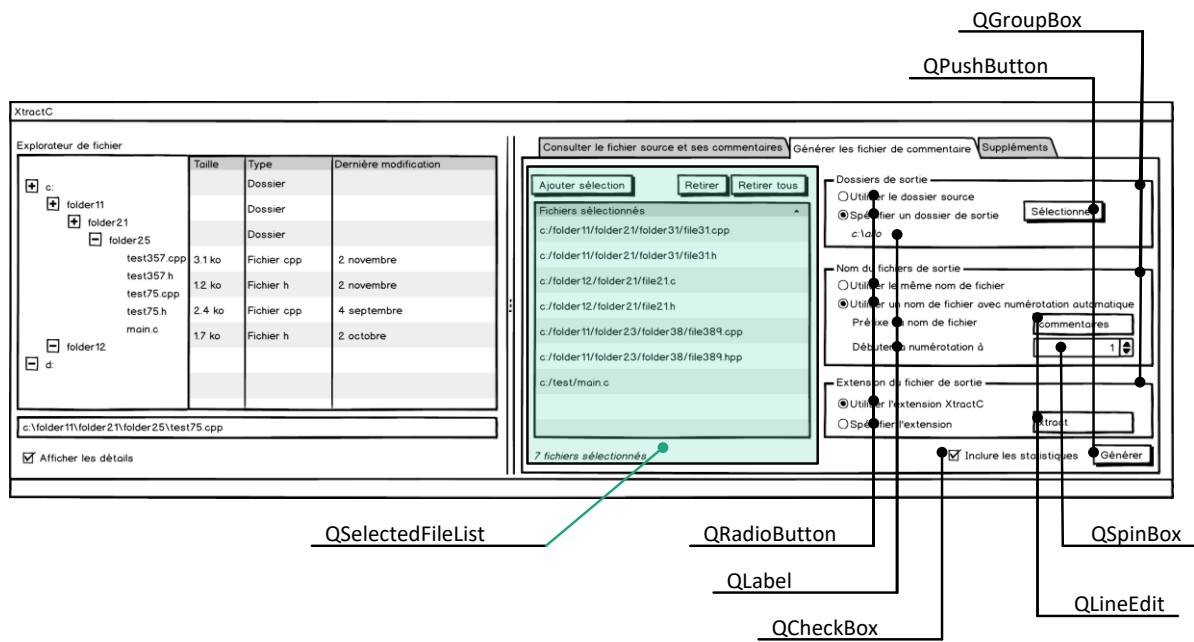
## 8.2 QDIRECTORYSELECTOR



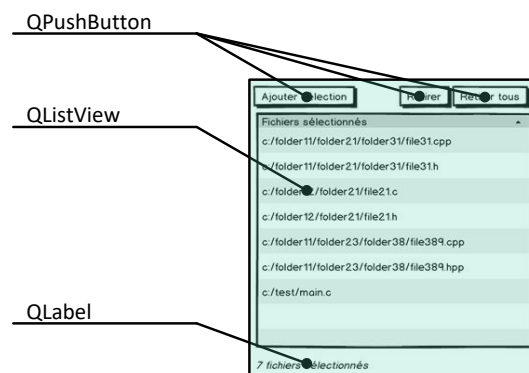
## 8.3 QCPPCOMMENTVIEWER



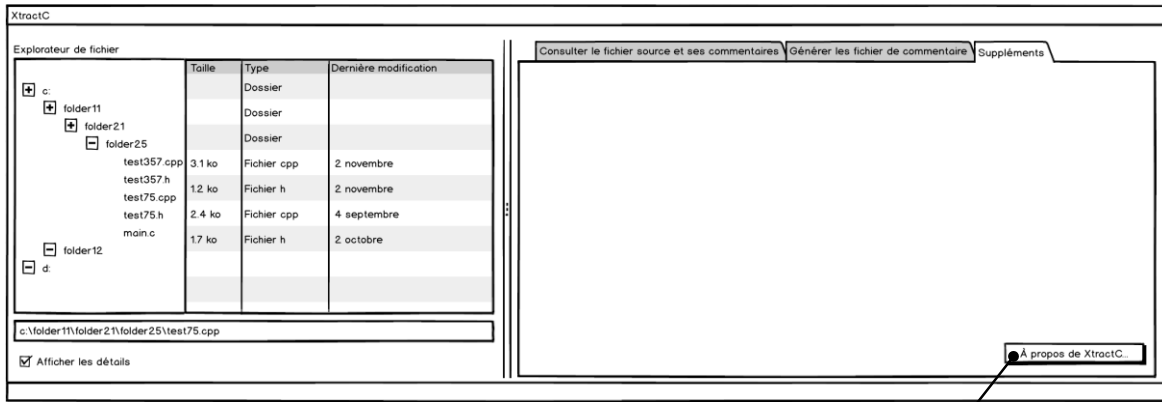
## 8.4 OUTIL DE GÉNÉRATION DE FICHIERS DE SORTIE



## 8.5 QSELECTEDFILELIST



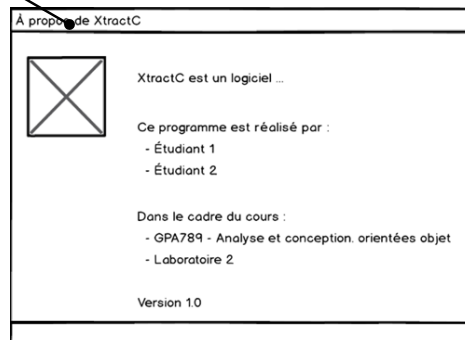
## 8.6 SUPPLÉMENTS



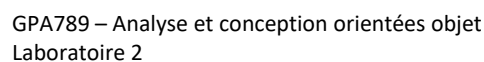
QPushButton

## 8.7 BOÎTE DE DIALOGUE À PROPOS

QMessageBox



## 9.1 EXPLORATEUR DE FICHIERS ET CONSULTATION



## 9.2 OUTIL DE GÉNÉRATION DE FICHIERS DE SORTIE

The screenshot displays the XtractC application interface. On the left is a file explorer showing the directory structure of the Qt5.7.0 source code. The center pane lists source files with columns for Name, Size, Type, and Date. The right pane contains configuration options for generating output files.

**Explorateur de fichiers**

Nom	Taille	Type	Dernier
OS (C:)		Drive	2016-
Apps		File Folder	2016-
DELL		File Folder	2016-
Facing		File Folder	2016-
Intel		File Folder	2016-
PerfLogs		File Folder	2016-
Program Files		File Folder	2016-
Program Files (x86)		File Folder	2016-
Qt		File Folder	2016-
Qt5.7.0		File Folder	2016-
5.7		File Folder	2016-
msvc2015_64		File Folder	2016-
bin		File Folder	2016-
doc		File Folder	2016-
include		File Folder	2016-
ActiveQt		File Folder	2016-
Qt3DCore		File Folder	2016-
Qt3DExtras		File Folder	2016-
Qt3DInput		File Folder	2016-
Qt3DLogic		File Folder	2016-
Qt3DQuick		File Folder	2016-
Qt3DQuickExtras		File Folder	2016-
Qt3DQuickInput		File Folder	2016-
Qt3DQuickRender		File Folder	2016-
Qt3DRender		File Folder	2016-
QtANGLE		File Folder	2016-
QtBluetooth		File Folder	2016-
QtCharts		File Folder	2016-
QtClucene		File Folder	2016-
QtConcurrent		File Folder	2016-
QtCore		File Folder	2016-
5.7.0		File Folder	2016-
Q_PID	23 bytes	File	1983-
Q_SECURITY_ATTRIBUTES	23 bytes	File	1983-
Q_STARTUPINFO	23 bytes	File	1983-
QAbstractAnimation	33 bytes	File	1983-
qabstractanimation.h	5 KB	h File	1983-
QAbstractEventDispatcher	39 bytes	File	1983-
qabstracteventdispatcher.h	4 KB	h File	1983-
QAbstractItemModel	33 bytes	File	1983-
qabstractitemmodel.h	19 KB	h File	1983-
QAbstractListModel	33 bytes	File	1983-
qabstractlistmodel.h	41 bytes	File	1983-
QAbstractNativeEventFilter	41 bytes	File	1983-
qabstractnativeeventfilter.h	2 KB	h File	1983-
QAbstractProxyModel	34 bytes	File	1983-
qabstractproxymodel.h	5 KB	h File	1983-
QAbstractState	29 bytes	File	1983-

**Consulter le fichier source et ses commentaires** | **Générer les fichiers commentaires** | **Suppléments**

**Ajouter sélection** | **Retirer** | **Retirer tout**

Dossier(s) de sortie  
☒ Utiliser le dossier source  
☐ Spécifier un dossier de sortie :

Nom du fichier de sortie  
☒ Utiliser le même nom de fichier  
☐ Utiliser un nom de fichier avec numérotation automatique  
Préfix du nom de fichier :   
Débuter la numérotation à :

Extension du fichier de sortie  
☒ Utiliser l'extension : xtractC  
☐ Spécifier l'extension :

18 fichiers sélectionnés

☒ Inclure les statistiques

**Générer**

## 9.3 SUPPLÉMENTS ET BOÎTE DE DIALOGUE À PROPOS

