

Plan intermédiaire

L3 Info. CDAA

R. Raffin

Graphiques

Framework
Graphics View

QPainter

Gestion
d'événements
souris

QPicture et
QImage

Base de données

JSON

Doxygen

Internationalisa-
tion

Machine à état

Architecture
Modèle-Vue

- 14 Graphiques
- 15 Framework Graphics View
- 16 QPainter
- 17 Gestion d'événements souris
- 18 QPicture et QImage
- 19 Base de données
- 20 JSON
- 21 Doxygen
- 22 Internationalisation
- 23 Machine à état
- 24 Architecture Modèle-Vue



Introduction

L3 Info. CDAA

R. Raffin

Graphiques

Framework
Graphics View

QPainter

Gestion
d'événements
souris

QPicture et
QImage

Base de données

JSON

Doxygen

Internationalisa-
tion

Machine à état

Architecture
Modèle-Vue

Qt intègre son propre système de traduction. La prise en charge de plusieurs langues est extrêmement simple dans les applications et ajoute peu de surcharge de travail pour le programmeur.

Selon le manuel de Qt Linguist, l'internationalisation est assurée par la collaboration de trois types de personnes : les développeurs, le chef de projet et les traducteurs.



Fig. : Exemple de rendu Qt Linguist



Intégration

L3 Info. CDAA

R. Raffin

Graphiques

Framework
Graphics View

QPainter

Gestion
d'événements
souris

QPicture et
QImage

Base de données

JSON

Doxygen

Internationalisa-
tion

Machine à état

Architecture
Modèle-Vue

Dans leur code source, les développeurs entrent des chaînes de caractères dans leur propre langue. Ils doivent permettre la traduction de ces chaînes grâce à la méthode `tr()`.

En cas d'ambiguïté sur le sens d'une expression, ils peuvent également indiquer des commentaires destinés à aider les traducteurs (2e champ du `tr()`).

Le chef de projet déclare les fichiers de traduction (un pour chaque langue) dans le fichier de projet (`.pro`) :

```
TRANSLATIONS += \  
    linguist0_fr_FR.ts \  
    linguist0_es_ES.ts \  
    linguist0_en_US.ts
```

L'utilitaire `lupdate` parcourt les sources à la recherche de chaînes à traduire et synchronise les fichiers de traduction avec les sources. Les fichiers de traductions sont des fichiers XML portant l'extension `.ts`.



Intégration

L3 Info. CDAA

R. Raffin

Graphiques

Framework
Graphics View

QPainter

Gestion
d'événements
souris

QPicture et
QImage

Base de données

JSON

Doxygen

Internationalisa-
tion

Machine à état

Architecture
Modèle-Vue

Les traducteurs utilisent Qt Linguist pour renseigner les fichiers de traduction.

Quand les traductions sont terminées, le chef de projet peut compiler les fichiers `.ts` à l'aide de l'utilitaire `lrelease` qui génère des fichiers binaires portant l'extension `.qm`, exploitables par le programme.

Ces fichiers sont lus à l'exécution et les chaînes de caractères qui y sont trouvées remplacent celles qui ont été écrites par les développeurs.



Processus d'intégration

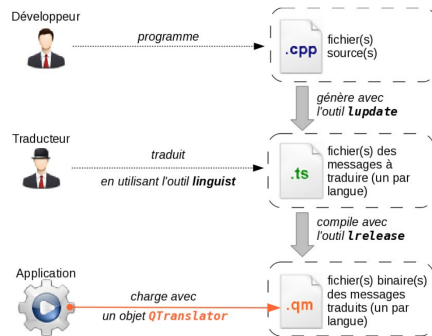


Fig. : Intégration QtLinguist

Étapes

■ le développeur

```
1 hellotr.cpp
2 QPushButton hello(QPushButton::tr("
    Hello world!"));
```

■ le chef de projet

```
1 hellotr.pro
2 SOURCES = hellotr.cpp
3 TRANSLATIONS = hellotr_fr.ts
```

■ le traducteur

```
1 $ lupdate -verbose hellotr.pro
```

■ le chef de projet

```
1 $ linguist hellotr_fr.ts
```

```
1 $ lrelease hellotr_fr.ts
```

Le traducteur utilise Qt Linguist pour renseigner les fichiers de traduction.

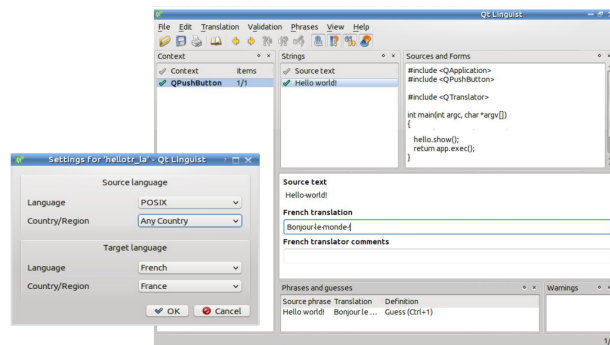


Fig. : Interface de Qt Linguist

Réalisation

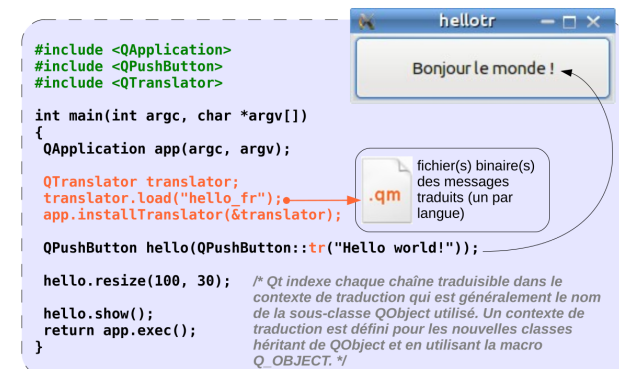


Fig. : Exemple

Application

L3 Info. CDAA

Internationalisa-
tion

Faisons un exemple :

- 1 création d'une QMainWindow simple avec une ui
- 2 un QPushButton avec un texte par défaut (traduisible)
- 3 un fonction qui contient un qDebug ()
- 4 le connect entre le bouton et la fonction

À chaque fois que l'on utilise une chaîne on place un `tr()` (pour dire « *traduisible* »).

Let's go...