

Zasoby, pliki graficzne

Bogdan Kreczmer

bogdan.kreczmer@pwr.wroc.pl

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki
Politechnika Wrocławska

Kurs: Wizualizacja danych sensorycznych

Copyright©2018 Bogdan Kreczmer

Niniejszy dokument zawiera materiały do wykładu dotyczącego programowania obiektowego. Jest on udostępniony pod warunkiem wykorzystania wyłącznie do własnych prywatnych potrzeb i może on być kopiowany wyłącznie w całości, razem z niniejszą stroną tytułową.

Niniejsza prezentacja została wykonana przy użyciu systemu składu \LaTeX oraz stylu beamer, którego autorem jest Till Tantau.

Strona domowa projektu Beamer:

<http://latex-beamer.sourceforge.net>

1 Zasoby

- Opis i dostęp do zasobów
- Lokalizacja zasobów
- Użycie zasobów w programie
- Użycie zasobów w bibliotece

Opis zasobów

Zasoby skojarzone z daną aplikacją muszą znajdować się w pliku z rozszerzeniem *.qrc. Zawiera on opis w formacie XML dostępnych zasobów.

Podawana ścieżka dostępu do pliku w opisie zasobów jest ścieżką względem kartoteki, w której znajduje się plik *.qrc.

```
<!DOCTYPE RCC><RCC version="1.0">  
  <qresource>  
    <file>img/led/led_blue.png</file>  
    <file>img/led/led_green.png</file>  
    <file>img/led/led_lightGray.png</file>  
    <file>img/led/led_red.png</file>  
    <file>img/led/led_yellow.png</file>  
  </qresource>  
</RCC>
```

Lokalizacja zasobów

Zasoby mogą być skompilowane do binarnej formy i dostępne bezpośrednio w kodzie. Mogą również zostać w formie binarnej i *zarejestrowane* później przez system zarządzający zasobami.

Dostęp do zasobów

Ścieżka względna	-	:/img/led/led_green.png
URL	-	qrc:///img/led/led_green.png

Możliwe jest tworzenie skrótowych odwołań poprzez aliasy

```
<file alias="LED-green.png">img/led/led_green.png</file>
```

Dostęp do zasobów

Ścieżka względna	-	:/LED-green.png
URL	-	qrc:///LED-green.png

Wiele sekcji

W danym pliku może być kilka sekcji opisujących zasoby. Można dodać prefiksy, które rozróżnią poszczególne sekcje.

```
<!DOCTYPE RCC> <RCC version="1.0">  
  <qresource prefix="LED">  
    <file alias="green.png">img/led_green.png</file>  
  </qresource>  
  <qresource prefix="icons">  
    <file>img/file.png</file>  
  </qresource>  
</RCC>
```

Dostęp do zasobów

```
:/LED/img/led_green.png  
qrc:///LED/img/led_green.png  
  
:/LED/green.png  
:/icons/img/file.png
```

Zróżnicowanie ze względu na lokalizację

Tworzenie różnych sekcji może być wykorzystane do umieszczania zasobów, które będą zróżnicowane ze względu na używany język i lokalizację.

```
<qresource>
  <file>flaga.png</file>
</qresource>
<qresource lang="en">
  <file alias="flaga.png">img/flag_en.png</file>
</qresource>
```

Dostęp do zasobów

```
:/flaga.png
qrc:///flaga.png
```

Wkompilowywanie zasobów

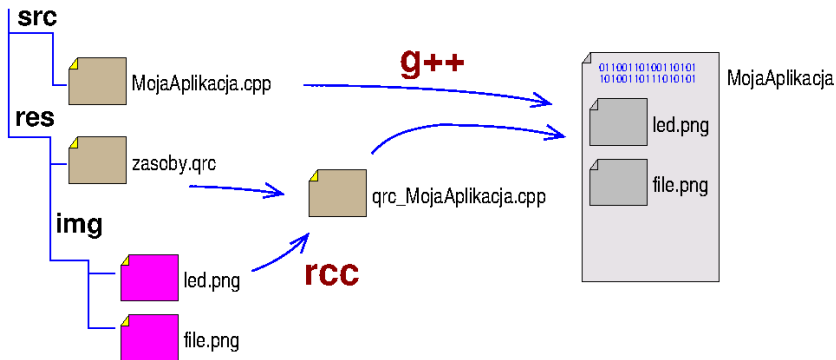
Wszystkie wykorzystywane zasoby możemy umieścić w kodzie aplikacji. Dzięki temu nie będzie ona wymagała dodatkowej informacji o lokalizacji zasobów. Aby to było możliwe, w projekcie aplikacji (plik *.pro) musi się pojawić odpowiedni wpis.

Wpis w pliku *.pro

```
RESOURCES = application.qrc
```

Dodatkowy wpis powoduje utworzenie odpowiednich reguł w pliku Makefile (w dostarczanych przykładach jest to plik Makefile.app), które wymuszają uruchomienie programu rcc. Narzędzie to tworzy moduł `qrc_NazwaProgramu.cpp`, w którym umieszczane są odpowiednie mapy bitowe w formacie XPM.

Przebieg przetwarzania zasobów



Kompresja zasobów

Program `rcc` domyślnie kompresuje zasoby. Można jednak ją wyłączyć wykorzystując parametr `-no-compress`. To może być użyteczne, gdy dołączamy zasoby, które już są skompresowane, np. pliki `*.png`

Wywołanie program `rcc`

```
rcc -no-compress zasoby.qrc
```

Można też kontrolować poziom kompresji wykorzystując parametry `-compress` i `-threshold`

Wywołanie program `rcc`

```
rcc -compress 2 -threshold 3 zasoby.qrc
```

Parametry wywołania programu rcc

Jeżeli korzystamy z programu `qmake` do zbudowania wszystkich reguł (jest to najwygodniejsze), to parametry wywołania programu `rcc` można zadawać poprzez zmienną `QMAKE_RESOURCE_FLAGS`.

Wywołanie program `rcc`

```
QMAKE_RESOURCE_FLAGS += -compress 2 -threshold 3
```

Inne zmienn programy `qmake` dla ustawień `rcc`

`RCC_DIR` – kartoteka, w której znajdują się produkty kompilacji programu `rcc`.
`RESOURCES` – zbiór plików `.qrc`

Odwołanie się do zasobów

Z zasobów graficznych można skorzystać w obiektach takich klas jak QIcon, QImage, QPixmap.

Przykład użycia

```
QLabel Etykieta;  
Etykieta.setPixmap(QPixmap(":/flaga.png"));
```

W pamięci komputera zasoby są przechowywane w strukturze drzewiastej. Struktura ta jest budowana automatycznie w momencie startu aplikacji. Do lokalizacji zasobu w tej strukturze wykorzystywane są obiekty QFile. Można też wykorzystać obiekt QDir. Inicjalizując go ścieżką "://" można przeglądać i nawigować po całej strukturze drzewiastej zasobów.

System zasobów wspiera również mechanizm zbioru automatycznie przeszukiwanych ścieżek. Odpowiednie ścieżki można dodawać wykorzystując metodę QDir::addSearchPath().

Użycie zasobów wewnątrz biblioteki

Gdy zasoby są wykorzystywane w module bibliotecznym konieczne jest ich zainicjalizowanie. Należy to zrobić przed ich pierwszym użyciem.

```
KlasaWBibliotece::KlasaWBibliotece() : KlasaBazowa()  
{  
    Q_INIT_RESOURCE(zasoby);    // zasoby.qrc  
  
    QFile Plik(":/flaga.dat");  
    ...  
}
```

Użycie zasobów biblioteki w programie głównym

Gdy zasoby biblioteki są wykorzystywane w programie głównym, to należy je tam zainicjalizować.

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    Q_INIT_RESOURCE(graphlib);

    QFile file(":/flaga.png");
    ...
    return app.exec();
}
```

Ten typ inicjalizacji może być również wykonany w przypadku bibliotek dynamicznie ładowanych (plugin'ów). Gdy są one zwalniane, zasoby należy jawnie usunąć wywołując polecenie `Q_CLEANUP_RESOURCE()`.

Koniec prezentacji
Dziękuję za uwagę