# Graficzny interfejs użytkownika --- szkic do wklejenia i pocięcia

Wstęp

Przeprowadzanie jakichkolwiek testów z udziałem losowych osób wymaga, aby testerzy byli prowadzeni przez cały test niejako za rękę. Wiążę się to z przemyśleniem i zaprojektowaniem interfejsu użytkownika w taki sposób, aby zapewnić możliwie jak największą czytelność i ergonomie procesu testowania. Interfejs użytkownika musi pozwalać na oglądanie filmów i wystawianie ocen, do interfejsu można wprowadzić dodatkowe panele pozwalające na informowanie testera, czy też pobieranie od niego dodatkowych informacji.

Narzędzia

*QT* to zestaw bibliotek dedykowanych dla m.in. języka C++ pozwalający na tworzenie zaawansowanych interfejsów użytkownika. *QT* zawiera również elementy pozwalające na obsługę procesów, sieci i grafiki trójwymiarowej a także integrację z bazami danych, posiada także narzędzia pozwalające na przeprowadzenie lokalizacji dla innych wersji językowych programu. Wykorzystane narzędzia zostały szczegółowo omówione w części pracy poświęconej graficznemu interfejsowi użytkownika. Wybór bibliotek *QT* jako narzędzia do tworzenia *GUI* (*graphic user interface*) był niejako oczywisty ze względu na fakt, iż VLC wykorzystuje jako QT jako jeden z podstawowych interfejsów użytkownika w odtwarzaczu podstawowym. Kluczowa była także opinia Jean-Baptiste Kempfa jednego z autorów i moderatorów biblioteki VLC, a także prezydenta i administratora forum jej poświęconego. Po przeanalizowaniu innych dostępnych możliwości uznano, iż jest to jedyny dostępny produkt dający możliwość swobodnego tworzenia paneli np. do oceny obejrzanego filmu.

QT Creator jedno z narzędzi udostępnianych w ramach zestawu bibliotek QT. Jest to kolejne użyte w pracy środowisko programistyczne pozwalające nie tylko na edycje kodu źródłowego, ale także (w narzędziu QT Designer) na projektowanie graficznego interfejsu użytkowania w okienkowym edytorze za pomocą widżetów z biblioteki QT. Za pomocą dodatkowych narzędzi uic i moc każda klasa korzystająca z sygnałów i slotów QT (reprezentująca element GUI) otrzymuje dodatkowe pliki \*.cpp i \*.ui reprezentujące rozmieszczenia okien, ustawienia grafiki, standardowe zdarzenia i przypisane do nich metody.

VLC-QT darmowa biblioteka autorstwa Tadej Novaka służąca do połączenia bibliotek QT z biblioteką libvlc. Pozwala ona na stworzenie prostego odtwarzacza wraz z dowolnym interfejsem pozwalającym na kontrolowanie odtwarzania. VLC-QT nakrywa metody i klasy upraszczając użycie libvlc w oknach QT. Ze względu na ograniczenia spowodowane przez użycie interfejsu imem do wczytywanie klatek nieskompresowanych sekwencji wideo, użycie VLC-QT zostało ograniczone do użycia widżetu wideo dającego większe możliwości niż standardowe QFrame.

Konfiguracja środowiska

Przed przystąpieniem do pracy z bibliotekami QT należy skonfigurować środowisko programistyczne. Ponieważ projekt został stworzony w środowisku CLion opartym na cmake należało zmodyfikować plik skrypt konfiguracyjny, dodają do linkera ścieżki do uprzednio zainstalowanej biblioteki QT. Dla poprawnego działania narzędzi biblioteki, a także do automatycznej generacji koniecznych do kompilacji plików \*.ui oraz ui\_\*.h należy w skrypcie CMakeLists uruchomić następujące parametry:



Podstawowe połączenie vlc z qt

Odtwarzacz wideo libvlc\_media\_player posiada domyślny interfejs (w zależności od ustawień może być oparty o QT) jednakże interfejsu tego nie można w żaden sposób modyfikować. Dlatego też do modyfikowania interfejsu odtwarzacza należy stworzyć własny widżet wideo. Aby to uczynić w najbardziej podstawowy sposób należy utworzyć klasę dziedziczącą z QWidget, a następnie utworzyć w niej obiekt typu QFrame, który następnie należy połączyć z odtwarzaczem za pomocą metody libvlc\_media\_player\_set\_xwindow(), która za argumenty przyjmuje utworzoną wcześniej instancje libvlc\_media\_player, a także identyfikator okna, który uzyskujemy wołając metodę winId() na obiekcie okna. Następnie można uruchamiać i kończyć wideo tak samo jak robiono to dotychczas bez dodatkowego interfejsu, pamiętając o tym, że nie wolno zniszczyć obiektu okna, do którego przypisany jest odtwarzacz wideo. Standardowy obiekt QFrame możemy modyfikować nadając mu zależne od nas wymiary, czy też umieszczając go w layoucie w wybranym przez siebie miejscu otaczając dowolnymi obiektami dodając np. paski do sterowania głośnością. Jednak ze względu na konieczność otwierania i zamykanie widżetu wideo, a także stworzenia paneli do oceniania i prowadzenia testera zastosowano inne podejście.

Zaproponowany Interfejs

Wstęp

Ze względu na silny związek interfejsu użytkownika ze scenariuszami testowymi (końcowa wersja powinna korespondować z opracowanymi scenariuszami pozwalając na ich przeprowadzenie) postanowiono zaprojektować interfejsy dedykowane do konkretnych scenariuszy dopiero po ich określeniu. Ponieważ zdecydowano się na standardowe metody ACR, PC oraz menu z wyborem filmów postanowiono opracować GUI oparte o modułowe panele podmieniane automatycznie w zależności od wybranej konfiguracji testów.

Projekt Gui

Stworzony Interfejs