Detecting when a Qt application has been switched to the background and when resumed

This code snippet shows how to create an event filter in Qt C++ to detect when an app has been switched to the background and when it is resumed. Note that for apps using Qt Quick Components for Symbian, the Symbian Element foreground property can be used instead.

Introduction

In many Qt supported platforms it is possible to switch the application to the background and resume to it when needed. For example in Harmattan the applications can be swiped to the background, and in Symbian^3 the switching can be done via the menu button. It is common that applications need to run some operations when these kinds of deactivation and activation events happen. For example games usually need to be paused when they are switched to the background. Also in other kinds of applications the unnecessary operations can be paused to free processing and memory resources.

Summary

In Qt this can be handled by listening the `ApplicationDeactivate` and `ApplicationActivate` events and reacting to them accordingly. The `QObject` class has an `eventFilter` method, which can be overridden in some class derived from `QObject`. Those earlier mentioned events can be listened in this reimplemented method. The class that reimplements the `eventFilter` method is then installed as an event filter for the QApplication object.

This methodology is used in Qt GameEnabler.

Sources

eventfilter.cpp

EventFilter derived from QObject

```
bool EventFilter::eventFilter(QObject *obj, QEvent *event)
{
   if (event->type() == QEvent::ApplicationDeactivate) {
      // The application deactivation can be handled here
      return true; // The event is handled
   }
   if (event->type() == QEvent::ApplicationActivate) {
      // The application activation can be handled here
      return true;
   }
   return QObject::eventFilter(obj, event); // Unhandled events are passed to the base
class
}
```

main.cpp

```
#include "eventfilter.h"

int main(int argc, char *argv[])
{
   QApplication app(argc, argv);
   EventFilter eventFilter;
   app.installEventFilter(&eventFilter); // Installing the event filter
   ...
}
```


(C) Copyright Nokia 2014. All rights reserved.

Postconditions

This code snippet demonstrated how to detect when a Qt application has been switched to the background and when it is resumed.

(C) Copyright Nokia 2014. All rights reserved.