


Intégration de SDL dans Qt

par Denys Bulant ([Tutoriels Qt](#))

Date de publication :

Dernière mise à jour :

Comment intégrer SDL dans un widget Qt ?

Ce tutoriel va explorer l'intégration de la bibliothèque  **SDL** dans Qt, le rendu étant effectué dans un widget Qt.


- I - Introduction
- II - Pré requis
- III - Un widget Qt comme cible de rendu pour SDL
 - III-A - Signifier à SDL la fenêtre à utiliser
 - III-B - Configuration du widget de rendu
 - III-C - Conflit de main
- IV - Modification du fichier projet
- V - Exemple complet
- VI - Annexes

I - Introduction

Ce tutoriel va explorer l'intégration de la bibliothèque  **SDL** dans Qt, le rendu étant effectué dans un widget Qt.

Version : Qt4 (Qt3 possible avec quelques ajustements).

II - Pré requis

- La  **librairie SDL** : le site contient une section documentation pour débiter.

III - Un widget Qt comme cible de rendu pour SDL

Les moteurs graphiques des différents Qt ne sont souvent pas adaptés au travail graphique intense (sauf à passer par OpenGL, mais on détourne le problème - c'est d'ailleurs ce que l'on va faire ici).

III-A - Signifier à SDL la fenêtre à utiliser

Se servir d'un widget comme zone de rendu cible pour SDL est très simple: il suffit d'une variable d'environnement.

Cette variable est `SDL_WINDOWID` et peut être de 2 formats :


- Un entier, tel quel ;
- Une valeur hexadécimale, préfixée par '0x'.

Une variable valide est un handle de fenêtre (dont le format peut varier en fonction de l'environnement utilisé). Ce handle s'obtient par la méthode `QWidget::winId()`, qui est une méthode publique.

Une fois la zone de rendu définie par cette variable, on peut initialiser le moteur graphique de SDL.

Voici par exemple le code permettant de réaliser cette intégration :

```
char windowid[64];
#ifdef Q_WS_WIN
    sprintf(windowid, "SDL_WINDOWID=0x%x", reinterpret_cast<qlonglong>(winId()));
#elif defined Q_WS_X11
    sprintf(windowid, "SDL_WINDOWID=0x%x", winId());
#else
    qFatal("Fatal: cast du winId() inconnu pour votre plate-forme; toute information est la bienvenue!");
#endif
SDL_putenv(windowid);
// Initialisation du système vidéo de SDL
SDL_Init(SDL_INIT_VIDEO);
screen = SDL_SetVideoMode(width(), height(), 32, SDL_SWSURFACE);
```

 *Ne le faites pas avant l'initialisation du système vidéo de SDL et de QWidget, sinon une seconde fenêtre sera créée par et pour SDL.*

III-B - Configuration du widget de rendu

Le dessin direct par SDL sur un widget Qt implique une petite configuration des attributs de dessin du dit widget :

```
setAttribute(Qt::WA_PaintOnScreen);
setAttribute(Qt::WA_NoSystemBackground);
```

À faire lors de l'utilisation de SDL, sinon, vous aurez une bouillie de pixels "inspirée" par le fond de l'écran à cet emplacement ;)

III-C - Conflit de main

Il y a par contre un conflit entre SDL et Qt au niveau de la fonction **main** . Lorsque **main** n'est pas le point d'entrée d'un programme (ce qui est le cas sur **win32** par exemple), Qt appelle **main** à partir de son implémentation de **WinMain** . SDL, quant à lui, fait un **define** :

Code extrait de `SDL_main.h`

```
#define main SDL_main
/* The prototype for the application's main() function */
extern C_LINKAGE int SDL_main(int argc, char *argv[]);
```

Pour supprimer ce problème, il vous faut annuler la définition de SDL juste après l'inclusion de **SDL.h** :

```
#include "SDL.h"
#undef main
```

Une fois ceci fait, vous pouvez utiliser sereinement le système vidéo de SDL au sein de votre application Qt :)

IV - Modification du fichier projet

Comme toute librairie, son utilisation avec votre programme doit être expressément signalée sous peine d'erreurs de compilation et/ou de liaison.

Les informations à ajouter dans le fichier projet sont :

- Le chemin vers les en-têtes SDL (ex. : **INCLUDEPATH += /usr/local/include/**) ;
- La librairie avec laquelle lier (et éventuellement son chemin) (ex. : **LIBS += -L/usr/local/lib/ -ISDL**).

V - Exemple complet

Vous trouverez associé à ce tutoriel un exemple illustrant l'utilisation de la SDL pour un starfield - qui rappellera des souvenirs aux (ex-)possesseurs de vieilles machines et qui ont codé avant l'arrivée du S-VGA (défini comme étant haute-résolution à l'époque) ;-)

J'ai repris ici les points principaux de la classe :

```
[...]

#ifdef Q_WS_WIN
#include <SDL.h>
#elif defined Q_WS_X11
#include <SDL/SDL.h>
#endif

/* PIEGE: main est redéfini par SDL.h comme SDL_Main.
   On retire donc la définition puisque Qt le gère
*/
#undef main

[...]

class SDLWidget : public QWidget
{
    Q_OBJECT

public:
    SDLWidget()
        :refreshTimer(0), windowInitialized(false), screen(0), StarNumbers(100)
    {
        [...]
    }

    virtual ~SDLWidget()
    {
        SDL_Quit();
    }

protected:
    virtual void showEvent(QShowEvent *e)
    {
        if(!windowInitialized)
        {
            // C'est ici qu'on dis à SDL d'utiliser notre widget
            char windowid[64];
#ifdef Q_WS_WIN
            sprintf(windowid, "SDL_WINDOWID=0x%x", reinterpret_cast<qlonglong>(winId()));
#elif defined Q_WS_X11
            sprintf(windowid, "SDL_WINDOWID=0x%x", winId());
#else
            qFatal("Fatal: cast du winId() inconnu pour votre plate-forme; toute information est la
            bienvenue!");
#endif
            SDL_putenv(windowid);

            // Initialisation du système vidéo de SDL
            SDL_Init(SDL_INIT_VIDEO);
            screen = SDL_SetVideoMode(width(), height(), 32, SDL_SWSURFACE);
            windowInitialized = true;
        }
    }
}
```

```
private:
    [...]

private:
    [...]


private slots:
    void onRefresh()
    {
        if(windowInitialized && screen)
        {
            SDL_LockSurface(screen);
            // Nettoyage de l'écran
            SDL_FillRect(screen, NULL, 0);
            // Dessin du starfield
            drawStarfield();
            SDL_UnlockSurface(screen);

            // Rafraichissement...
            SDL_UpdateRect(screen, 0, 0, 0, 0);

            // Et enfin, mise à jour des positions des étoiles
            updateStarfield();
        }
    }

private:
    [...]
};
```

N'oubliez pas d'ajouter les références d'inclusion et de bibliothèques de SDL au **.pro** , et laissez-vous aller à un peu de nostalgie ;-)

L'article original est situé ici :  [Intégration de SDL](#) .

VI - Annexes

Source [FTP](#) ou [HTTP](#) .

