

# 7

## Créer des interfaces avec Qt Quick Designer

---

Le langage QML permet d'élaborer une interface graphique indifféremment par le biais d'un outil d'édition graphique ou en [éditant des fichiers texte](#). Le présent chapitre se propose de vous guider dans la conception d'une interface graphique en utilisant principalement l'éditeur graphique inclus dans Qt Creator, le *Qt Quick Designer*, de manière élégante et simple.

Nous avons scindé le chapitre en deux parties, de complexité croissante. La première, basée sur le Designer, cible en premier lieu les infographistes et designers. Nous y avons intentionnellement fait abstraction du code QML généré par l'éditeur. En conséquence, cette partie est accessible à quiconque, sans connaissances prérequis en programmation.

L'approche basée sur le Qt Quick Designer est adaptée au développeur novice mais n'empêche en rien des scénarios de développement plus avancés. La seconde partie, plus ambitieuse, se propose d'expliquer comment étendre son interface en éditant le code source. Le programmeur et le designer enthousiaste y découvriront comment exercer un contrôle direct des éléments mis en place à l'aide du Designer.

Cette dualité permet de séparer proprement le code du design de l'interface utilisateur et autorise un gain de productivité, dans la mesure où les diverses tâches sont effectuées à l'aide des outils qui leur sont plus adaptés.

### Séparation du code source et de l'interface

Il est tentant de choisir et d'assembler des contrôles au gré de l'écriture du code source. Cependant cette approche souffre d'un écueil : il devient de plus en plus difficile de maintenir et de faire évoluer le code, au fur et à mesure de l'avancée du projet.

Le développement efficace d'une interface graphique exige méthode et cohérence. Dans un souci de clarté et de maintenabilité, il est recommandé de séparer autant que possible le code source et l'interface graphique, de sorte que la modification

de l'un n'ait qu'un impact minime sur l'autre. Le contenu de ce chapitre se fonde sur ce paradigme.

Un autre avantage de la séparation scrupuleuse du code et de l'interface graphique est que le design peut être confié à une personne dont la création graphique est la spécialité, sans qu'il ait besoin de s'initier à la programmation auparavant. Il pourra se consacrer à loisir, et jusque dans le moindre détail, à l'aspect visuel de l'interface, à la charte graphique et à l'expérience utilisateur.

Bien sûr, une bonne communication entre le designer et le développeur est d'autant plus avantageuse que le projet devient ambitieux, voire primordiale lorsque par exemple une nouvelle fonctionnalité requise par le designer n'a pas encore été implémentée par le développeur.

## 7.1. Présentation du projet

Pour vous présenter Qt Quick Designer, nous allons nous appuyer sur un exemple concret : la conception, en partant de zéro, de l'interface graphique d'un outil de mixage audio. Si vous n'avez jamais utilisé un tel outil, ne soyez pas effrayé ! Nous allons vous expliquer de manière succincte de quoi il est constitué.

*Note* > Dans le jargon d'un technicien du son, mixer est un terme désignant le fait de mélanger les signaux provenant de plusieurs sources distinctes (les voies) en un seul signal (le master). Par exemple, lors de l'enregistrement d'un morceau de musique, les différents instruments et les voix des différents chanteurs sont typiquement enregistrés individuellement puis mixés en un morceau final. L'intérêt est que l'on peut à tout moment retravailler à volonté la composition, donner plus ou moins d'importance à l'une ou l'autre source et ajuster le timbre d'un instrument en réglant ses paramètres. L'outil dont il est question ici est une version rudimentaire d'une table de mixage.

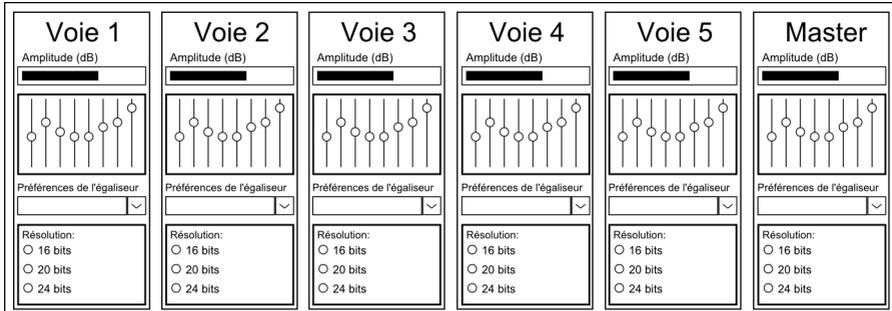
### Cahier des charges

L'outil que nous souhaitons mettre en place doit permettre d'effectuer divers réglages sur des signaux audio provenant de différentes sources, constituant ce que l'on nomme des voies. Une configuration typique consistera par exemple à consacrer trois voies pour la guitare, le synthétiseur et la batterie, plus une voie pour le chanteur et une autre pour les chœurs. Les signaux audio résultant de chacune des voies, après réglages individuels, sont combinés en un signal unique, nommé le *master*. Le master dispose lui-même de son propre module de réglage. Certains réglages sont spécifiques à chaque voie, tandis que d'autres sont globaux et affectent l'ensemble des voies.

Nous souhaitons un module de réglage par voie, qui sera répété à l'identique pour le nombre de voies disponibles et pour le master. La disposition des modules est similaire à

celle d'une console de mixage réelle, à savoir une juxtaposition verticale des éléments de réglage en une colonne. Les modules des voies sont placés côte à côte et le module du master est placé tout à droite. La [Figure 7.1](#) présente l'esquisse qui sert de base à la mise en place de l'interface.

**Figure 7.1 :** Esquisse de l'interface graphique de l'exemple



## 7.2. Prendre en main son espace de travail

Qt Quick Designer regroupe au sein d'une unique interface tous les outils dont le développeur d'une application peut avoir besoin. La vaste palette d'outils mis à votre disposition peut paraître intimidante au premier abord. Cette section a pour but de démystifier cette apparente complexité et de détailler les principaux constituants dont vous pourriez être amené à avoir besoin.

Pour commencer, nous allons configurer les outils du Designer de manière à pouvoir travailler de manière efficace.

**Note** > Les réglages décrits dans cette section ne sont accessibles, pour la plupart, que lorsqu'un projet est ouvert. Pour suivre les explications et expérimenter par vous-même dans Qt Creator, nous vous suggérons de créer d'ores et déjà un nouveau projet de type QT QUICK 2 UI WITH CONTROLS en suivant les instructions données à la [section Créer un nouveau projet](#).

### Configurer les panneaux latéraux

De part et d'autres de la fenêtre centrale se situent deux panneaux configurables dont l'affichage est optionnel. Leur affichage peut être activé ou désactivé par le biais du menu FENÊTRE > VUES > BASCULER PANNEAU LATÉRAL GAUCHE (ou par le biais du raccourci clavier Ctrl+Alt+0) et BASCULER PANNEAU LATÉRAL DROIT (ou par le biais du raccourci clavier Ctrl+Alt+Maj+0).

Si vous avez besoin temporairement d'une vue plus large de la zone de dessin, vous pouvez désactiver et réactiver à souhait les panneaux latéraux droit et gauche en basculant la vue pleine écran à l'aide du raccourci clavier Alt+0.

Chaque panneau est configurable et peut contenir de un à cinq éléments au choix parmi :

- le navigateur des contrôles de l'interface ;
- la bibliothèque des éléments disponibles ;
- la liste des documents ouverts ;
- l'éditeur de propriétés ;
- le navigateur de projets ;
- le navigateur de fichiers.

Par défaut, le panneau gauche contient le NAVIGATEUR DES CONTRÔLES DE L'INTERFACE et la BIBLIOTHÈQUE DES ÉLÉMENTS DISPONIBLES, tandis que le panneau de droite contient l'ÉDITEUR DE PROPRIÉTÉS.

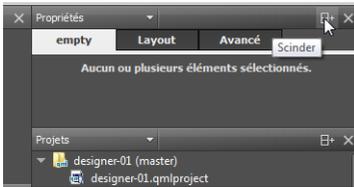
Un menu accessible dans le coin supérieur droit de chaque élément offre les options SCINDER pour rajouter un élément dans le même panneau ou FERMER pour supprimer un élément du panneau. Vous pouvez à tout moment revenir à la configuration par défaut par le biais du menu FENÊTRE > VUES > RESTAURER LA VUE PAR DÉFAUT.

*Astuce > Si vous souhaitez optimiser votre outil de création, vous gagnerez à tester les différentes configurations possibles jusqu'à avoir trouvé la combinaison qui vous convient le mieux. La configuration proposée ici ne l'est qu'à titre indicatif.*

Le panneau gauche contient deux éléments importants, dont nous nous servirons régulièrement. Il n'y a donc pas lieu de le modifier. L'élément PROPRIÉTÉS du panneau de droite convient aussi. Nous rajouterons en dessous de celui-ci le panneau du NAVIGATEUR DE PROJETS en cliquant sur le symbole SCINDER dans le coin supérieur droit de la colonne de droite.

Trois types de navigateur de fichiers peuvent être utilisés. Le premier permet de lister tous les fichiers ouverts dans Qt Creator. Le deuxième liste les projets ouverts dans Qt Creator et les fichiers leur appartenant. Le dernier, enfin, affiche l'arborescence des fichiers tels qu'ils apparaissent dans le système de fichiers. Nous choisissons le navigateur de projets, qui se révélera utile pour passer rapidement d'un fichier QML à un autre lorsque le projet sera plus étoffé. La configuration obtenue est représentée sur la [Figure 7.2](#).

Figure 7.2 : Navigateur de projets



## Le navigateur de l'interface

Ce navigateur présente de manière hiérarchique l'ensemble des éléments de l'interface contenus dans la zone de dessin. Ces deux représentations (celle du navigateur et celle de la zone de dessin) sont étroitement liées de sorte que toute modification d'un élément dans l'une se répercute sur l'autre.

La présentation sous forme d'arborescence met en évidence les liens de parenté des éléments. Pour chacun d'entre eux, vous pouvez réassigner les liens de parenté (à l'aide des icônes en forme de flèches ou par glisser-déposer) et activer ou désactiver son affichage (à l'aide de l'icône en forme d'œil).

*Note > Les panneaux configurables peuvent être redimensionnés librement. En fonction de la largeur de celles-ci, une partie ou la totalité des flèches mentionnées peuvent être masquées. Dans ce cas, elles sont accessibles uniquement par le biais des chevrons >> situés dans le coin supérieur droit du panneau.*

## Prévisualiser la scène QML

Vous serez fréquemment amené à vouloir vérifier le rendu d'une vue QML telle qu'elle sera perçue par l'utilisateur final. Pour exécuter le projet actif et visualiser votre interface QML, vous pouvez utiliser la commande EXÉCUTER, représentée par l'icône en forme de triangle vert ▶ associée au raccourci de clavier Ctrl+R.

## 7.3. Structurer le projet de manière modulaire

Comme vous pouvez le voir sur la [Figure 7.1](#), l'interface graphique que nous voulons réaliser est constituée de six parties semblables, correspondant à autant de voies, dont les entêtes respectifs sont Voie 1 à Voie 5 et Master. Elle a donc ceci de particulier qu'elle consiste en une répétition à l'identique d'un module de base, à raison d'un par voie. Effectuer le travail six fois de suite (ou reproduire les éléments par copier-coller) peut se révéler plus simple à court terme mais a pour conséquence de multiplier les risques d'erreur. En outre, si vous souhaitez ensuite modifier le moindre élément sur un