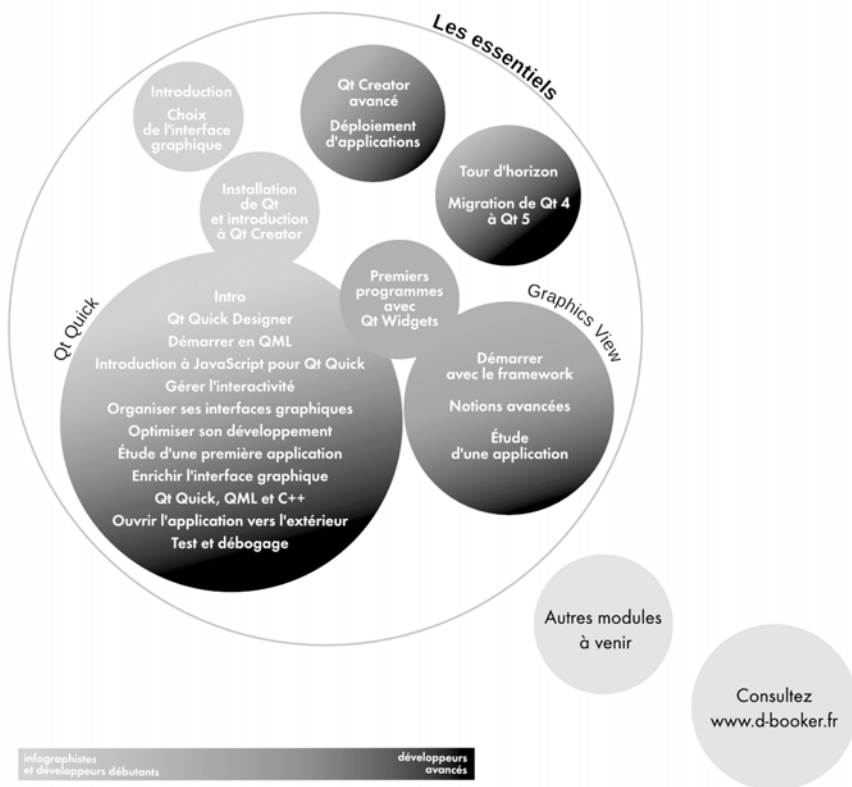


# Aperçu général de l'ouvrage

Créer des applications avec Qt 5 a été conçu selon une logique thématique plutôt qu'une progression linéaire. Il est constitué d'un ensemble de modules, pouvant être lus dans l'ordre que vous voulez. Bien sûr, si vous débutez, commencez par les chapitres correspondant à votre niveau.



# Table des matières

---

Préface .....	xiii
Introduction .....	xv
1. Qt, pourquoi faire ?.....	xv
2. Objectifs du livre .....	xvi
3. Contenu du livre .....	xvi
4. Par où commencer ?.....	xvii
5. Prérequis .....	xix
6. Codes sources des exemples .....	xx
7. Accès aux vidéos .....	xx
Choix de l'interface graphique .....	xxi
<b>1. Tour d'horizon .....</b>	<b>1</b>
1.1. Les nouveautés de Qt 5 .....	2
Organisation générale .....	2
QPA .....	4
Qt Quick 2 .....	5
Qt Scene Graph .....	7
WebKit2 .....	8
Qt 3D .....	9
Utilisation de C++11 .....	9
Divers .....	10
1.2. Les plateformes ciblées par Qt 5 .....	10
1.3. Les niveaux de maturité .....	11
<b>2. Migration de Qt 4 à Qt 5 .....</b>	<b>15</b>
2.1. Stratégie de migration .....	15
2.2. Modification des sources C++ .....	16
Éléments supprimés .....	17
Réorganisation des modules .....	21
Changements d'API incompatibles avec Qt 4 .....	22
Chargement des plug-ins .....	25
Changements de comportement .....	26
Dépréciations .....	29
2.3. Gestion des projets .....	30
qmake .....	30
CMake .....	31
QBS .....	33

2.4. Spécificités des plateformes .....	33
Linux .....	33
Mac OS .....	33
Windows .....	34
Symbian et MeeGo .....	34
<b>3. Installation de Qt et introduction à Qt Creator.....</b>	<b>35</b>
3.1. Installation de Qt .....	35
Par l'installateur .....	36
Par les dépôts .....	38
Par une compilation .....	39
Comparaison des méthodes .....	47
3.2. Introduction à Qt Creator .....	48
Pourquoi Qt Creator .....	48
Première utilisation .....	49
<b>4. Premiers programmes (avec Qt Widgets).....</b>	<b>55</b>
4.1. Bonjour Qt .....	55
4.2. Signaux .....	58
4.3. Client pour un service web .....	60
Enginio .....	60
Spécifications .....	61
Préparation du projet .....	61
Construction de l'interface utilisateur .....	62
Traitements dans l'interface utilisateur .....	68
Communication avec le serveur .....	69
Fin de l'application .....	74
Traductions .....	74
Tests .....	76
Déploiement .....	80
Aller plus loin .....	81
<b>5. Aller plus loin avec Qt Creator .....</b>	<b>83</b>
5.1. Mise en place et configuration du projet .....	84
Gestion des Kits .....	90
5.2. Intégration du versionning .....	92
5.3. Les sessions Qt Creator .....	96
5.4. QWidget et le prototypage rapide dans Qt Creator .....	97
Édition des widgets .....	99
Édition des signaux & slots .....	100
Édition des copains .....	101

Édition de l'ordre des tabulations .....	102
Prévisualisation des interfaces .ui .....	102
5.5. Le lanceur d'outils externes .....	103
5.6. Navigation avancée .....	105
Modes et panneaux de sortie .....	106
Navigation dans et entre les fichiers du projet .....	106
Localisateur .....	109
5.7. Profiling de code .....	110
<b>6. Déploiement d'applications .....</b>	<b>113</b>
6.1. Bibliothèques statiques et partagées .....	113
6.2. Déploiement de Qt .....	116
6.3. Spécificités de Windows .....	118
6.4. Outils de déploiement .....	118
<b>Qt Quick.....</b>	<b>121</b>
<b>Introduction .....</b>	<b>123</b>
1. Qu'est-ce que Qt Quick et le QML ?.....	123
2. Organisation de ce module du livre .....	125
3. Premiers pas avec Qt Quick.....	126
Créer un nouveau projet .....	126
Modifier le projet via l'éditeur de code .....	128
Modifier le projet via l'éditeur graphique .....	129
<b>7. Créer des interfaces avec Qt Quick Designer.....</b>	<b>131</b>
7.1. Présentation du projet .....	132
Cahier des charges .....	132
7.2. Prendre en main son espace de travail .....	133
Configurer les panneaux latéraux .....	133
Le navigateur de l'interface .....	135
Prévisualiser la scène QML .....	135
7.3. Structurer le projet de manière modulaire .....	135
Réutilisation d'éléments de base .....	136
7.4. Mettre en place l'interface pour une voie .....	136
Créer un projet .....	137
Configurer le projet .....	137
Choisir ses éléments dans la bibliothèque .....	138
Positionner les éléments de l'interface .....	138
7.5. Convertir l'interface en un composant .....	145

7.6. Réutiliser le composant .....	147
7.7. États et transitions .....	149
États .....	150
Transitions .....	152
<b>8. Démarrer en QML .....</b>	<b>155</b>
8.1. Concepts d'objet, héritage et composition .....	155
8.2. Premier document QML .....	157
Importer un module .....	158
Créer un élément .....	159
Ajouter des commentaires .....	162
8.3. Attribuer des identifiants aux éléments .....	162
8.4. Paramétrer les éléments .....	164
Initialiser une propriété .....	164
Types des propriétés .....	165
Lier des propriétés .....	179
8.5. Fonctionnalités de base des éléments graphiques .....	180
Afficher ou masquer un élément .....	181
Contrôler l'ordre d'affichage avec <i>z-order</i> .....	184
Transformations, anti-aliasing et cliping .....	185
8.6. Contrôles de base .....	192
8.7. Afficher du texte .....	195
Mise en forme du texte .....	195
Ajouter un lien dans le texte .....	201
Aligner le texte .....	202
Élision et retour à la ligne automatique .....	202
Régler l'interlignage .....	205
8.8. Afficher une image .....	206
<b>9. Introduction à JavaScript pour Qt Quick.....</b>	<b>211</b>
9.1. JavaScript et ECMAScript .....	211
9.2. Éléments de syntaxe .....	213
Variables .....	213
Types .....	214
Commentaires .....	215
Opérateurs .....	215
Fonctions .....	216
9.3. Programmation impérative .....	218
Conditions .....	218
Boucles .....	222

9.4. Utilisation du code JavaScript dans du code QML .....	224
9.5. Spécificités de l'environnement Qt Quick.....	226
Restrictions .....	226
État .....	227
Importation .....	229
Inclusion .....	229
9.6. Comparaison de JavaScript avec C++ .....	230
9.7. JavaScript avancé .....	231
Comparaison sur le type .....	231
Programmation fonctionnelle .....	231
Programmation orienté objet .....	234
Objet global .....	237
<b>10. Gérer l'interactivité .....</b>	<b>239</b>
10.1. Un concept de base : le <i>signal handler</i> .....	239
10.2. Gérer les événements de la souris .....	240
Modifier l'aspect du curseur lors du survol d'un élément .....	240
Modifier l'aspect du curseur lors d'un clic de souris .....	241
Modifier l'aspect d'un élément lors du survol ou d'un clic .....	243
Récupérer la position de la souris pour déplacer un élément. .	245
Créer un bouton cliquable .....	246
Compter le nombre de clics de souris .....	246
Compter le nombre de clics de souris sur des liens HTML .....	247
Détecter la modification d'une propriété .....	249
Déplacer un élément avec le glisser-déposer .....	250
Déposer un élément sur un autre avec le glisser-déposer .....	251
10.3. Permettre la saisie d'informations .....	254
Champs de saisie pour identifiant et mot de passe .....	255
Fonctionnalités de base des champs de saisie .....	256
Mettre le champ de saisie actif en avant .....	257
Supprimer automatiquement le texte par défaut .....	258
Masquer le mot de passe .....	259
Limiter la largeur de l'affichage du texte saisi .....	260
À propos du défilement automatique du texte saisi .....	261
Imposer des restrictions sur le contenu d'un champ de saisie. .	262
10.4. Créer un éditeur de texte simple .....	269
Créer un champ d'édition .....	269
Utiliser les fonctions de sélection du texte .....	270
Modifier le texte d'un champ d'édition .....	273
10.5. Gérer les événements du clavier .....	276

Propagation des événements du clavier .....	276
Activer un élément lors du lancement de l'application .....	279
Modifier dynamiquement le focus .....	280
Utiliser la touche tabulation pour changer le focus .....	280
Utiliser les flèches du clavier pour changer le focus .....	281
Gérer n'importe quelle touche du clavier .....	282
10.6. Aller plus loin avec les signaux .....	284
Propriétés et <i>signal handlers</i> attachés .....	285
10.7. Annexe .....	286
Toutes les propriétés de <i>MouseArea</i> .....	286
Tous les signaux handlers de <i>MouseArea</i> .....	287
Comprendre les informations liées à un signal .....	288
Drag, <i>DropArea</i> et <i>DragEvent</i> .....	289
<b>11. Organiser ses interfaces graphiques.....</b>	<b>291</b>
11.1. Créer une fenêtre principale .....	291
Fenêtre simple .....	292
Fenêtre secondaire .....	294
Fenêtre avec masque .....	298
Fenêtre d'application .....	298
Gérer l'écran .....	306
11.2. Disposer les éléments dans une page .....	308
Positionner manuellement des éléments graphiques .....	308
Positionner avec des ancres .....	310
Organiser en ligne, en colonne ou dans une grille .....	313
Positionner les éléments les uns à la suite des autres .....	332
11.3. Organiser et afficher les pages .....	335
<b>12. Optimiser son développement .....</b>	<b>339</b>
12.1. Créer des composants réutilisables en QML .....	339
Quelques exemples .....	340
Lancer une fonction à la création/destruction d'un élément.....	341
Définir un composant QML réutilisable .....	342
Définir l'interface d'un composant réutilisable .....	346
Créer dynamiquement des éléments en QML .....	352
12.2. Utiliser des modèles et des vues .....	359
Les différents modèles .....	360
Choisir une vue adaptée .....	369
12.3. Ajouter des états et des transitions .....	381
Définir plusieurs états pour un objet .....	382

Créer des transitions .....	389
<b>13. Étude d'une première application avec Qt Quick .....</b>	<b>393</b>
13.1. Première étape : un prototype statique .....	395
13.2. Deuxième étape : boutons et actions .....	399
13.3. Troisième étape : affichage du résultat .....	401
13.4. Quatrième étape : historique .....	406
13.5. Cinquième étape : utilisation du clavier .....	411
13.6. Sixième étape : utilisation de Qt Quick Layouts.....	417
13.7. Septième étape : support d'autres opérations .....	423
13.8. Mot de la fin .....	424
Comparaison avec KCalc .....	425
<b>14. Enrichir l'interface graphique .....</b>	<b>427</b>
14.1. Dessiner en 2D avec Qt Quick.....	427
La surface de dessin, le système de coordonnées et le	
contexte .....	428
Couleurs .....	429
Tracé de chemins et remplissage .....	431
Extrémités et connexions de lignes .....	433
Dégradés et ombres .....	434
Images et motifs .....	436
14.2. Qt Quick Particles.....	439
Mise en œuvre .....	439
14.3. Animations .....	441
Animation directe des propriétés d'un élément .....	441
<b>15. Qt Quick, QML et le C++.....</b>	<b>443</b>
15.1. Application C++ et scène QML .....	444
QtQuick2ApplicationViewer .....	444
QQuickView .....	445
Moteur QML .....	447
Qt Widgets.....	449
15.2. Fonctionnement interne du moteur d'exécution .....	450
Création de la hiérarchie .....	450
Liaison entre les propriétés .....	451
Gestion des types .....	452
15.3. Communication entre C++ et QML .....	453
Création de propriétés contextuelles .....	453
Création de nouveaux types .....	460
15.4. Création d'un plug-in C++ pour QML .....	477



<b>16. Ouvrir l'application vers l'extérieur .....</b>	<b>483</b>
16.1. Ouvrir l'application vers le réseau .....	483
La transparence réseau de Qt Quick.....	483
Accéder à un service web avec le QML .....	490
16.2. Ouvrir l'application vers une base de données .....	504
Communication avec une base de données locale .....	504
Communication avec une base de données distante .....	509
<b>17. Tests et débogage .....</b>	<b>525</b>
17.1. Systématiser des tests .....	526
Des tests pour chaque situation .....	526
Module Qt Quick Test.....	527
Tests unitaires .....	531
Tests fonctionnels .....	535
17.2. Déboguer .....	542
Activation du débogage des projets Qt Quick.....	542
Débogage JavaScript .....	544
Débogage QML .....	547
17.3. Analyser les performances .....	549
Profilage .....	551
Tests d'évaluation des performances .....	554
<b>Graphics View .....</b>	<b>557</b>
<b>18. Démarrer avec le framework Graphics View .....</b>	<b>559</b>
18.1. Un outil d'une grande flexibilité .....	559
Son fonctionnement .....	560
Avantages & Contraintes .....	561
18.2. Premiers pas avec le framework .....	562
Architecture de la scène .....	562
Prise en main du système de coordonnées .....	563
Première mise en œuvre .....	565
18.3. Les QGraphicsItem, composants fondamentaux de la scène..	568
Utilisation des QGraphicsItem génériques .....	568
Créer ses propres QGraphicsItem .....	574
<b>19. Notions avancées .....</b>	<b>577</b>
19.1. Transformer une scène ou des éléments .....	577
Transformations simples .....	577
L'objet QTransform .....	578

QGraphicsTransform .....	580
19.2. Animer les éléments de la scène .....	581
Animations simples .....	581
Groupes d'animations .....	584
19.3. Interagir avec l'utilisateur .....	585
Les événements de souris .....	586
Les événements de clavier .....	587
La sélection .....	587
Le survol de la souris .....	590
Le glisser-déposer .....	592
19.4. Z-index et événements de souris .....	598
19.5. Intégrer des widgets .....	601
19.6. Polices de caractères, rendu des textes .....	604
19.7. Booster les performances de la QGraphicsView .....	605
L'indexation des éléments dans la scène .....	606
Types de mises à jour du viewport de la vue .....	607
Mise en cache de l'arrière-plan de la vue .....	608
Détails des flags d'optimisation .....	608
Mise en cache du QGraphicsItem .....	609
<b>20. Étude d'une application créée avec Graphics View.....</b>	<b>611</b>
20.1. Présentation de l'application .....	611
20.2. Structure générale .....	612
20.3. Programme principal .....	612
20.4. Fenêtre principale de l'application .....	613
20.5. Les données : <i>FileData</i> et <i>DirData</i> .....	614
20.6. Écran de chargement .....	618
20.7. La vue principale .....	621
La scène graphique .....	622
La classe View .....	625
Les éléments de la vue : <i>FileItem</i> et <i>DirItem</i> .....	628
Le widget de navigation .....	630
20.8. La fenêtre des propriétés de fichiers .....	632
20.9. Le graphe des types .....	633
Index général (hors Qt Quick) .....	637
Index Qt Quick .....	643
À propos des auteurs .....	649

# Préface

---

par **Nicolas Arnaud-Cormos**

*Senior Software Engineer, KDAB*

Qu'est-ce que Qt ? Il y a dix ans, j'aurais peut-être répondu rapidement sans vraiment réfléchir, en mettant en avant la simplicité d'utilisation, la documentation et le support multiplateforme. Ces qualités font et ont fait la renommée de Qt.

Mais en dix ans, le monde technologique a bien changé, les téléphones sont presque aussi puissants qu'un ordinateur, nombreux sont ceux ayant plusieurs terminaux connectés : ordinateurs, téléphones, tablettes, télévisions, montres, lunettes... avec des formes, des tailles et des interfaces différentes. Ce changement technologique s'est accompagné par une démocratisation du développement logiciel, permettant maintenant à l'amateur éclairé de créer et publier ses propres applications sur différents terminaux.

Cette évolution des terminaux et des usages s'accompagne également d'une véritable révolution de l'expérience utilisateur. Fini les applications avec menus et boutons classiques... l'utilisateur est maintenant habitué à des interfaces fluides et animées, presque "organiques". Les écrans tactiles ont maintenant permis l'utilisation de gestes, rendant leur utilisation plus naturelle. Il est par exemple devenu courant de zoomer sur une photo ou une page web en pinçant avec le pouce et l'index.

Dans ce nouveau contexte, Qt ne pouvait plus se satisfaire de ses acquis. Pour répondre à ces nouveaux besoins et usages, il a dû évoluer, innover et se réinventer, que ce soit au niveau du code, des performances mais aussi structurellement. Tous ces changements ont conduit à la parution de la version 5.0 en décembre 2012, et ce n'est que le début.

*Nobody knows what 'most' C++ programmers do* -Bjarne Stroustrup

C'est une leçon importante venant de l'inventeur même du C++, et qui peut aussi s'appliquer à Qt : la bibliothèque Qt ne doit pas présumer de son utilisation. Plutôt que de restreindre le développeur ou l'équipe de développement (bien souvent accompagnée d'un ou plusieurs designers), il faut au contraire les libérer de toutes contraintes. C'est un des grands chantiers techniques de ces dernières années, chantier qui a débuté dès Qt 4.7 avec l'apparition de QML et Qt Quick, et continue encore de nos jours.

De nombreux autres changements ont été opérés sur le support OpenGL, le support C++11, les performances... pour moi, un des changements les plus important est struc-

turel : le développement de la bibliothèque Qt est maintenant complètement ouvert. Développeurs et contributeurs utilisent tous les mêmes outils (il n'y a plus de différence entre contributions internes ou externes). Mais plus important encore : toutes les discussions et décisions sont maintenant publiques. Il est donc possible pour n'importe qui de participer et d'influencer librement le développement de la bibliothèque Qt.

Qu'est-ce que Qt ? C'est une question difficile dont la réponse dépend grandement de celui ou celle qui la pose. Pour l'un ce sera la communauté qui primera, pour un autre le côté multiplateforme, un troisième trouvera son bonheur dans la simplicité d'utilisation et de déploiement... Ce qui est sûr, c'est que Qt évolue, s'améliore et est disponible sur de plus en plus de plateformes (Linux, Mac OSX, Windows, Qnx, BlackBerry, Android, iOS...).

Quels que soient vos besoins, le présent livre va vous épauler dans votre découverte ou votre approfondissement de Qt. Les auteurs, plutôt que de vous proposer une lecture linéaire, vous incitent à venir piocher les chapitres qui vous intéressent, à creuser seulement certaines parties, selon votre bon vouloir.

Alors, apprenez, testez, survolez ou approfondissez, et surtout amusez-vous avec Qt, vous avez entre les mains l'ouvrage idéal pour vous accompagner !

# Introduction

---

## 1. Qt, pourquoi faire ?

Qt est un ensemble d'outils d'aide à la création de logiciels.

L'objectif de Qt est de faciliter le travail des développeurs de différentes manières :

- en étant multiplateforme : les applications créées avec Qt sont compatibles avec différents systèmes d'exploitation et le développeur peut se concentrer sur son applicatif plutôt que sur les détails spécifiques des plateformes visées ;
- en étant natif : les applications ne subissent pas de pénalité en terme de performances et d'intégration à la plateforme (ergonomie, design, accès au système) ;
- en étant complet : tous les ingrédients de base pour la réalisation d'applications sont intégrés, y compris des couches graphiques pour les interfaces utilisateurs.

En dehors de ces éléments quantitatifs, l'aspect qualitatif est également une composante très importante de Qt. Il s'agit du soin apporté aux API et à la documentation, et d'une réflexion permanente sur la "bonne façon" de faire les choses, en masquant autant que possible la complexité sous-jacente.

Le paysage applicatif est en évolution rapide depuis 2007, qui est l'année de la sortie du premier iPhone : les nouvelles plateformes mobiles (smartphones et tablettes) gagnent du terrain au détriment des plateformes bureautiques et des consoles de jeu ; un autre phénomène majeur est l'extension du domaine des applications web et des outils de développement associés. Sans être contradictoires, ces deux tendances ne sont pas convergentes : les besoins d'intégration au système, ainsi que les problèmes de continuité de la connexion internet rendent l'usage d'applications 100 % web difficile dans le contexte mobile.

Qt s'est adapté à ce paysage récent en proposant ses propres solutions aux nouveaux problèmes posés par les appareils mobiles et tactiles, non seulement en terme d'adaptation de l'interface utilisateur, mais aussi en terme d'adaptation aux cycles de développement de ces interfaces souvent beaucoup plus courts. Ces solutions sont basées essentiellement sur le module Qt Quick, qui sera présenté en détail dans le livre.

## 2. Objectifs du livre

Ce livre a pour ambition de vous présenter les outils de base de Qt 5, en mettant l'accent sur ses nouveautés. Le framework et ses usages étant très variés, il nous a semblé plus pertinent de les aborder de manière thématique plutôt que de se confiner dans une structure d'apprentissage progressive classique. Par ailleurs, nous souhaitons aussi bien intéresser le lecteur débutant en Qt que le développeur familier de Qt 4 désireux de se former à Qt 5. Il en résulte un ouvrage un peu particulier, organisé sous forme de modules plus ou moins autonomes – un peu à l'image du framework.

*Les essentiels* constituent une première série de modules, visant à introduire le framework dans son ensemble et les fonctionnalités Qt 5 les plus attendues. Par la force des choses, la partie traitant de Qt Quick et du QML est la plus importante : c'est là que résident les principales nouveautés de Qt 5 et un débutant devra nécessairement commencer par s'initier à leur emploi.

Si vous êtes un adepte de Qt Widgets, vous serez certainement un peu surpris du peu de place que nous lui avons accordé. Non pas que nous jugeons ce module majeur de Qt comme secondaire, mais du fait de sa présence historique au sein du framework, il est déjà très documenté. Nous avons préféré nous concentrer sur des aspects moins connus.

Malgré ses 600 pages, le présent livre a fait l'objet de choix draconiens. Et nous avons le sentiment de n'avoir abordé qu'une petite partie de Qt. C'est pourquoi, nous prévoyons de publier à l'avenir d'autres modules spécialisés, qui viendront compléter cette première base.

## 3. Contenu du livre

*Attention* > Ce livre n'est pas construit de manière linéaire. Si vous débutez, vous serez sans doute tenté de lire les chapitres dans l'ordre, or ce n'est pas forcément la meilleure façon ! Vous pourriez être un peu perdu. Commencez par les chapitres les plus débutants en vous aidant de la carte graphique de l'ouvrage et des parcours proposés à la Section 4, *Par où commencer ?*.

Vous trouverez dans *Les essentiels* les modules suivants :

- Tour d'horizon, qui introduit les nouveautés de Qt 5 au regard des versions antérieures, présente la philosophie de son développement et ses perspectives ;
- Migration de Qt 4 à Qt 5, qui comme son nom l'indique vous aidera à migrer. Ces deux premiers chapitres, de par leur nature, s'adressent plutôt aux développeurs déjà familiers de Qt ;
- Installation de Qt et introduction à Qt Creator, qui fournit aux plus débutants d'entre vous des informations relatives à l'installation du SDK et un premier aperçu

de Qt Creator, l'environnement de développement de référence. Vous y trouverez également des informations avancées pour des installations de Qt personnalisées ;

- Premiers programmes (avec Qt Widgets), qui passe en revue un ensemble de concepts importants de Qt 5 au travers de l'analyse de trois exemples. L'interface graphique de ces applications utilise le module Qt Widgets, ce qui peut constituer une introduction à ce module. Si vous avez déjà pratiqué Qt 4, vous y trouverez une illustration de fonctions nouvelles, ainsi qu'une petite introduction à Enginio, un service web disposant de fonctions d'intégration aux applications Qt ;
- Aller plus loin avec Qt Creator, qui présente différentes fonctions de Qt Creator fort utiles pour travailler efficacement et prendre en main l'environnement de développement dédié à Qt ;
- Qt Quick, qui vous initie en onze chapitres à l'utilisation de Qt Quick, le nouveau module de Qt permettant de développer facilement des interfaces graphiques en QML. Il se décompose en plusieurs chapitres allant du moins technique (à destination des graphistes et développeurs débutants) au plus technique (pour les développeurs C++).
- Graphics View, qui vous initie en trois chapitres (Les bases, notions avancées et exemple d'application) à l'emploi du framework QGraphicsView.

À ces modules s'ajoutent deux petits compléments : Choix de l'interface graphique qui vous aidera à choisir le module Qt dont vous avez besoin pour développer votre interface et Déploiement d'applications qui fournit quelques informations essentielles sur le déploiement d'une application avec Qt 5.

## 4. Par où commencer ?

Fondamentalement aucun ordre de lecture n'est imposé, vous pouvez lire les chapitres dans l'ordre que vous voulez, selon vos besoins et votre intérêt. Toutefois certains chapitres supposent que vous ayez acquis certaines connaissances de base, et d'autres s'adressent plus spécifiquement aux débutants. D'une manière générale, assurez-vous à l'aide de la carte graphique que le niveau du module qui vous intéresse corresponde bien à votre niveau.

### Parcours Débutants infographistes

Commencez par le chapitre Installation de Qt et introduction à Qt Creator en n'y lisant que la section portant sur l'installation de Qt par l'installateur et l'introduction à Qt Creator. Une fois Qt installé sur votre ordinateur, allez directement à la Partie Qt Quick, et commencez par la première partie du chapitre Créer des interfaces

avec Qt Quick Designer. Appropriiez-vous le mode Designer en créant vos propres exemples, et seulement après, si vous en sentez le besoin, essayez de comprendre ce qui se cache derrière vos objets : lisez l'introduction de la partie, puis le chapitre Démarrer en QML. Progressez ensuite dans l'ordre des chapitres aussi loin que vous suivez.

## **Parcours Débutants développeurs**

Commencez par le chapitre Installation de Qt et introduction à Qt Creator, sans entrer dans les détails de la compilation, puis allez directement à la Partie Qt Quick, lisez l'introduction puis passez directement à Démarrer en QML. Progressez ensuite dans l'ordre des chapitres ou attaquez directement par le chapitre Étude d'une première application avec Qt Quick en vous référant aux chapitres plus théoriques au fur et à mesure de sa lecture pour approfondir votre connaissance des outils utilisés. Une fois les notions de base acquises, la découverte du Designer pourra vous faire gagner du temps. Après Qt Quick, étendez votre connaissance de Qt avec les autres modules.

## **Parcours Développeurs débutant en Qt**

Vous développez déjà en C++ et souhaitez prendre en main Qt, commencez par Choix de l'interface graphique et Premiers programmes (avec Qt Widgets), puis laissez-vous guider par votre curiosité et vos besoins. N'oubliez pas le chapitre Aller plus loin avec Qt Creator, qui vous aidera à configurer votre environnement de développement selon vos habitudes.

## **Parcours Développeurs familiers de Qt 4**

Commencez par le Tour d'horizon afin de repérer les nouveaux éléments de Qt 5 qui vous intéressent. Si vous devez faire migrer vos applications, poursuivez avec Migration de Qt 4 à Qt 5. Puis poursuivez votre lecture selon vos centres d'intérêts. N'oubliez pas le chapitre Aller plus loin avec Qt Creator, qui vous aidera à configurer votre environnement de développement selon vos habitudes. Ni le chapitre Créer des interfaces avec Qt Quick Designer qui apportera des pistes pour améliorer la collaboration entre designers et développeurs.



## 5. Prérequis

Le module Qt Quick ayant été conçu pour permettre à des non-développeurs de créer des interfaces en Qt, les premiers chapitres ayant trait à son emploi n'exigent aucun prérequis. C'est dans cette perspective qu'un chapitre spécifique introduisant JavaScript pour Qt Quick vous est proposé. Les autres parties du livre supposent que le lecteur possède des compétences de base en programmation C++ et en programmation orientée objet. Aucune connaissance préalable du framework Qt n'est nécessaire, cependant l'ouvrage est consacré à Qt 5 et ne couvre que partiellement les fondations de Qt 4.

Pour utiliser les exemples de ce livre, il vous faudra créer un environnement de développement en téléchargeant le framework Qt 5.x (de préférence Qt 5.1 ou plus), un environnement de développement intégré (EDI) et un compilateur C++. Le développement Qt est supporté sous Windows, Mac OS et Linux avec les composants cités ci-dessus. L'installateur Qt fournit Qt Creator pour chacune de ces plateformes comme EDI de base, la Section 3.2, Introduction à Qt Creator vous présente comment bien débiter avec cette application et le chapitre Aller plus loin avec Qt Creator vous permet d'aller plus loin. Il est également possible d'utiliser d'autres EDI comme Visual Studio sous Windows ou Xcode sous Mac OS.

Il vous faudra également un compilateur C++ pour compiler les exemples du livre et les applications que vous allez créer. Visual Studio Express 2010 [[http://d-booker.jo.my/qt-vs-express](http://d-booker jo my/qt-vs-express)] et 2012 [[http://d-booker.jo.my/qt-vs-express-for-win-desktop](http://d-booker jo my/qt-vs-express-for-win-desktop)] sont disponibles gratuitement pour Windows ou vous pouvez télécharger MinGW [[http://d-booker.jo.my/qt-mingw-dl](http://d-booker jo my/qt-mingw-dl)] pour utiliser le compilateur GCC sur cette même plateforme. Les packages Linux et Mac OS proposent généralement une version récente de GCC. Depuis la version Qt 5.0.1, un installateur du framework pour Windows inclut les binaires de MinGW dans sa version 32 bits qui contient GCC 4.7.2. Qt 5.1 est depuis livré avec MinGW 4.8.

Si vous désirez faire du développement mobile, vous n'avez pas besoin d'un téléphone pour développer des applications Qt Quick. Des simulateurs pour les systèmes MeeGo et MeeGo sont disponibles avec Qt Simulator inclus dans Qt Creator. Le support de Android et iOS sont également disponibles en Technology Preview dans Qt 5.1. Il est cependant possible de brancher votre téléphone possédant un de ces systèmes d'exploitation avec un câble USB et de tester vos applications créées avec Qt Creator directement sur votre téléphone. Pour le développement mobile, des outils supplémentaires, comme un SDK spécifique à la plateforme, peuvent être nécessaires. Reportez-vous au chapitre Installation de Qt et introduction à Qt Creator pour plus d'informations.

## 6. Codes sources des exemples

Vous trouverez sur la page [<http://d-booker.jp.my/qt5-les-essentiels>] du site des éditions D-BookeR consacrée au livre, onglet Compléments, une archive réunissant l'ensemble des codes sources des exemples du livre. Ceux-ci sont classés par projet et chapitre.

Toutefois pour vous simplifier l'accès à ces codes au fur et à mesure de votre lecture, nous les avons aussi hébergés sur Github dans un dossier dédié [<http://d-booker.jp.my/qt5-1-sources>].

## 7. Accès aux vidéos

La version numérique du livre contient quelques illustrations sous forme de vidéos .

Dans la version imprimée, un QRCode à droite de l'image vous renvoie vers la galerie en ligne des vidéos [<http://d-booker.jp.my/qt5-1-videos>].