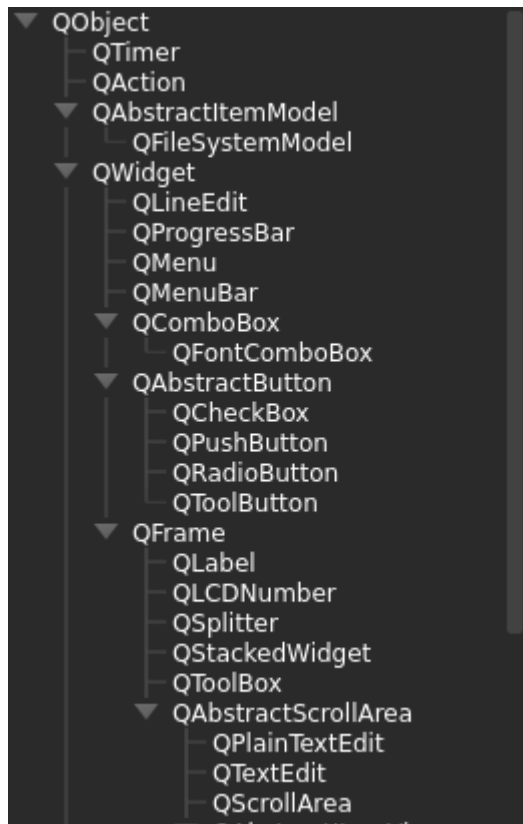


Qt-Engine

Qt-engine est une application permettant de concevoir une application sans une ligne de code. Un projet est créé dans lequel l'ont créé des "views". Chaque views représente une partie de l'application. Lorsque le projet est prêt à être testé, il peut être exporté et ainsi le code de l'application sera généré.

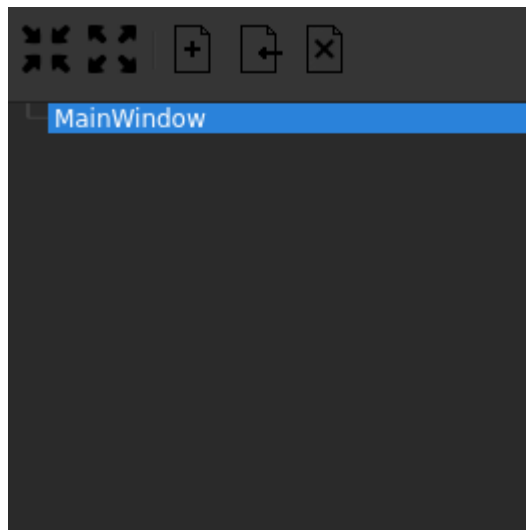
Qt-engine est muni d'un système de layout composé de divers panneaux. Il en existe actuellement 7 répertoriés ci-dessous. Chaque panneau peut s'assembler avec d'autres et se diviser. C'est ainsi qu'un layout peut être composé, sauvegarder et recharger par la suite.

I. Library Object



Ce panneau liste tous les objets supportés par Qt-Engine. Chaque objet peut être drag & drop de ce panneau à un autre panneau.

II. Project Explorer



Ce panneau permet de visualiser les différentes views. Un double-clic sur l'une d'elle permet de charger cette view.

On peut également apercevoir 5 boutons.

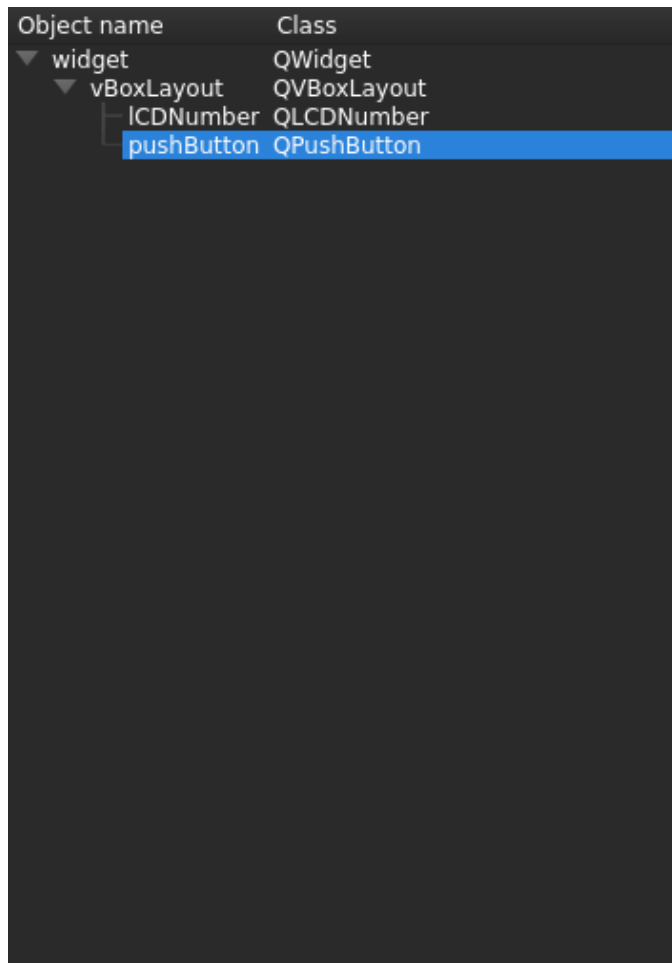
Le premier et le deuxième permettent respectivement d'étendre et de regrouper l'arbre des vues.

Le troisième à créer une nouvelle vue.

Le quatrième à importer une vue.

Le 5ième à supprimer une vue du projet.

III. View Explorer



Object name	Class
▼ widget	QWidget
▼ vboxLayout	QVBoxLayout
ICDNumber	QLCDNumber
pushButton	QPushButton

Ce panneau permet de visualiser les différents éléments d'une view.

Divisé en deux colonnes pour chaque ligne, la première colonne indique le nom de l'élément et la deuxième son type. Le nom de chaque élément peut être modifié en double cliquant sur la ligne correspondante.

Un élément peut être apporté de la librairie d'objet en effectuant un drag & drop. Il peut être positionné et repositionné à l'endroit voulue.

En effectuant un clic droit sur un élément, une liste d'action possible concernant l'élément en question est affichée.

En effectuant un clic gauche sur un élément, l'élément est sélectionné et devient l'élément courant.

IV. Object Property

Property	Value
▶ QObject	
▶ QWidget	
▼ QAbstractButton	
text	Click Me !
▶ iconSize	16 x 16
shortcut	
checkable	False
checked	<input type="checkbox"/> False
autoRepeat	False
autoExclusive	<input type="checkbox"/> False
autoRepeatDelay	300
autoRepeatInterval	100
▼ QPushButton	
autoDefault	<input type="checkbox"/> False
default	False
flat	<input type="checkbox"/> False

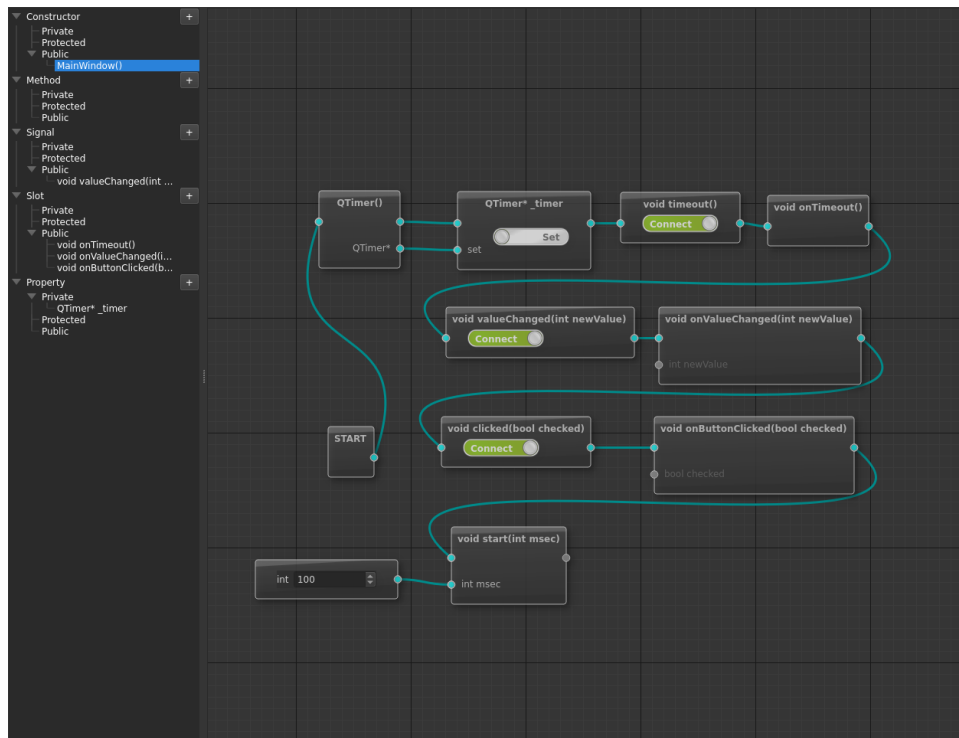
Ce panneau permet de visualiser et de modifier les propriétés de l'élément courant.

V. View



Ce panneau permet de visualiser la view actuellement chargée afin d'obtenir un aperçu de celle-ci. Certains éléments acceptent les interactions afin de modifier également les propriétés de cet élément.

VI. Workflow



Ce panneau est divisé en deux parties :

- La première à gauche permet de visualiser les constructeurs, méthodes, signaux, slots et propriétés créés par l'utilisateur. Il est donc possible d'en rajouter en cliquant sur le bouton "+" correspondant.

Par défaut est créé, et n'est ni modifiable, ni supprimable, un constructeur ne comprenant aucun argument.

Un clic droit sur une méthode ou autre (cité précédemment) permet d'afficher un menu permettant de modifier ce dernier ou de le supprimer. Un double clic sur un constructeur, slot ou méthode permet de visualiser et modifier son contenu dans la deuxième partie.

- La deuxième partie à droite permet de visualiser et modifier, comme dit précédemment, le contenu d'un constructeur, méthode ou slot.

La scène peut être déplacée lors d'un mouvement de souris tout en maintenant le bouton gauche de la souris.

Elle peut également être zoomée et dézoomée en utilisant la molette de la souris.

Un clic droit sur la scène permet d'ouvrir un menu proposant une liste de méthode, slot, signal et propriété utilisable pour la view actuellement chargée. La sélection de l'un d'eux génèrera sur la scène le "block" correspondant.

Un block peut être déplacé et relié à un autre à l'aide de ses pins. Deux pins sont reliables si et seulement si le type de ces derniers correspondent.

La scène accepte également des drops des panneaux suivants :

- **LibraryObject**, génère un block avec le constructeur par défaut de l'objet dropper.
- **ProjectExplorer**, génère un block avec le constructeur par défaut de la view dropper.
- **ViewExplorer**, ouvre un menu avec les méthodes, signaux, slots et propriétés disponibles de l'élément dropper.