

## Installation d'un environnement de développement pour MacOS:

Pour pouvoir suivre le cours, il vous faudra un environnement nécessaire au développement de projets C++ :

- une version suffisamment à jour du système d'exploitation (minimum Catalina 10.15) ;
- un compilateur C++ (obtenu via les command line tools) ;
- un environnement de développement intégré (EDI, conseillé: Qt Creator). Notez qu'un simple éditeur de texte peut suffire à la place d'un EDI (conseillé: Geany, dans ce cas), cependant cette dernière option peut s'avérer nettement moins confortable pour l'utilisation des outils de «debugging»;
- un outil de construction de projets : l'outil CMake (<https://cmake.org>) sera employé dans ce cours ;
- une librairie graphique : SFML (<https://www.sfml-dev.org/>) sera utilisée dans ce cours.

Cette page a pour but de vous guider dans le processus d'installation d'un tel environnement. Pour ce faire, nous allons d'abord installer Homebrew (<https://brew.sh/fr/>), un gestionnaire de paquets pour MacOS, qui facilitera le processus.

**Important: pour éviter des problèmes inutiles, il est fortement recommandé d'installer tous les composants décrits dans des répertoires ne comportant ni espace ni caractères spéciaux ou accentués dans leurs noms.**

### Installation de g++ (compilateur)

- Ouvrir le terminal (Application -> Utilitaires -> Terminal)
- Entrez :

```
xcode-select --install
```

Une fenêtre devrait s'ouvrir vous proposant d'installer Xcode, ou seulement les Command Lines Tools ; choisissez cette dernière option, puis « Install » et acceptez la licence.

- Attendez la fin de l'installation, puis entrer la commande :

```
g++ --version
```

Vous devriez alors voir s'afficher quelque chose de similaire à ceci (les numéros sont peut être différents) :

```
Apple clang version 14.0.3 (clang-1403.0.22.14.1)
Target: x86_64-apple-darwin22.5.0
```

```
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
```

ou ceci

```
Apple clang version 14.0.3 (clang-1403.0.22.14.1)
Target: x86_64-apple-darwin22.5.0
Thread model: posix
InstalledDir: /Applications/Xcode.app/Contents/Developer/Toolchains
/XcodeDefault.xctoolchain/usr/bin
```

## Installation de Homebrew

Pour installer Homebrew, exécutez la commande suivante dans un terminal :

```
/bin/bash -c
"$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

EN UNE SEULE LIGNE (pas de retour à la ligne après le -c)

## Installation de CMake

CMake est un outil permettant notamment de construire un exécutable à partir d'un ensemble de fichiers C++ constituant un projet.

Pour installer cet outil, exécutez la commande suivante dans un terminal :

```
brew install cmake
```

## Installation de Qt Creator

- Télécharger l'installateur MacOS à l'adresse <https://www.qt.io/download-qt-installer-oss>.
- Lancer l'installation. Vous aurez besoin de vous authentifier ou de créer un compte pour accéder à l'installateur, utilisez une adresse mail valide puis authentifiez vous via l'installateur)

Welcome to the Qt online installer

**Welcome**

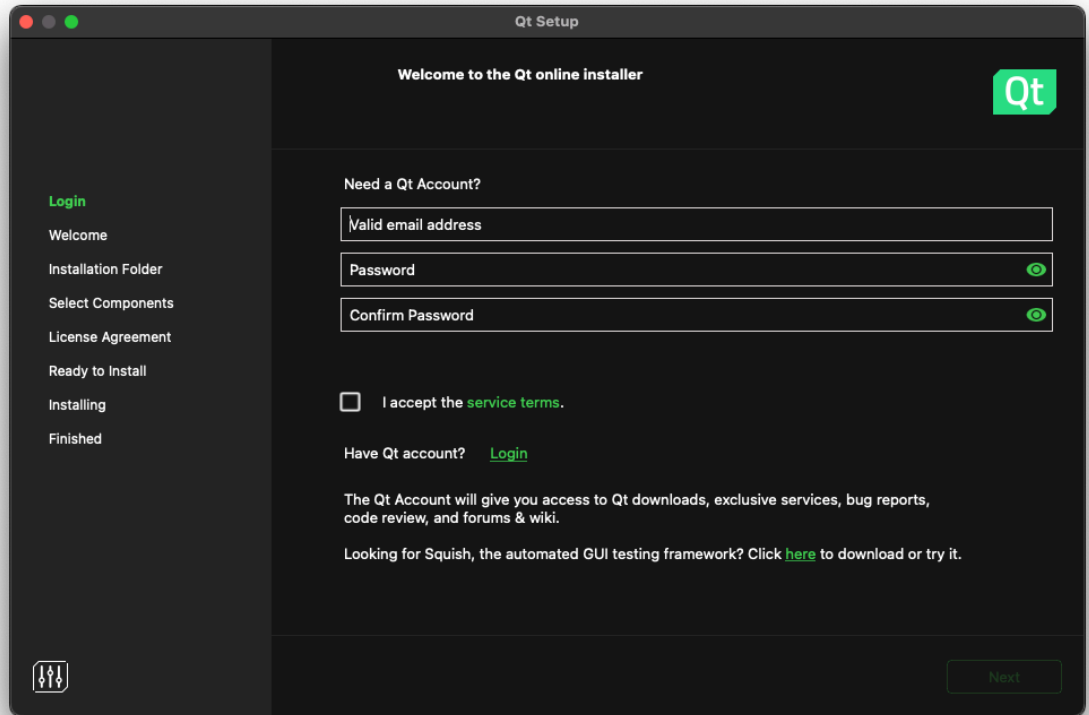
- Setup - Qt
- Installation Folder
- Select Components
- License Agreement
- Start Menu shortcuts
- Ready to Install
- Installing
- Finished

Please log in to Qt Account

[Forgot password?](#)

Don't have Qt account? [Sign up](#)

⏪ ⏩




Confirmation par courriel/création de compte : (Il suffit de remplir les champs Prénom et Nom de famille, de définir le troisième champ sur Éducateur/Étudiant et de cocher la première case)




## Your Account

First Name

Last Name

Educator/Student 

I am an individual person not using Qt for any company

Select country 

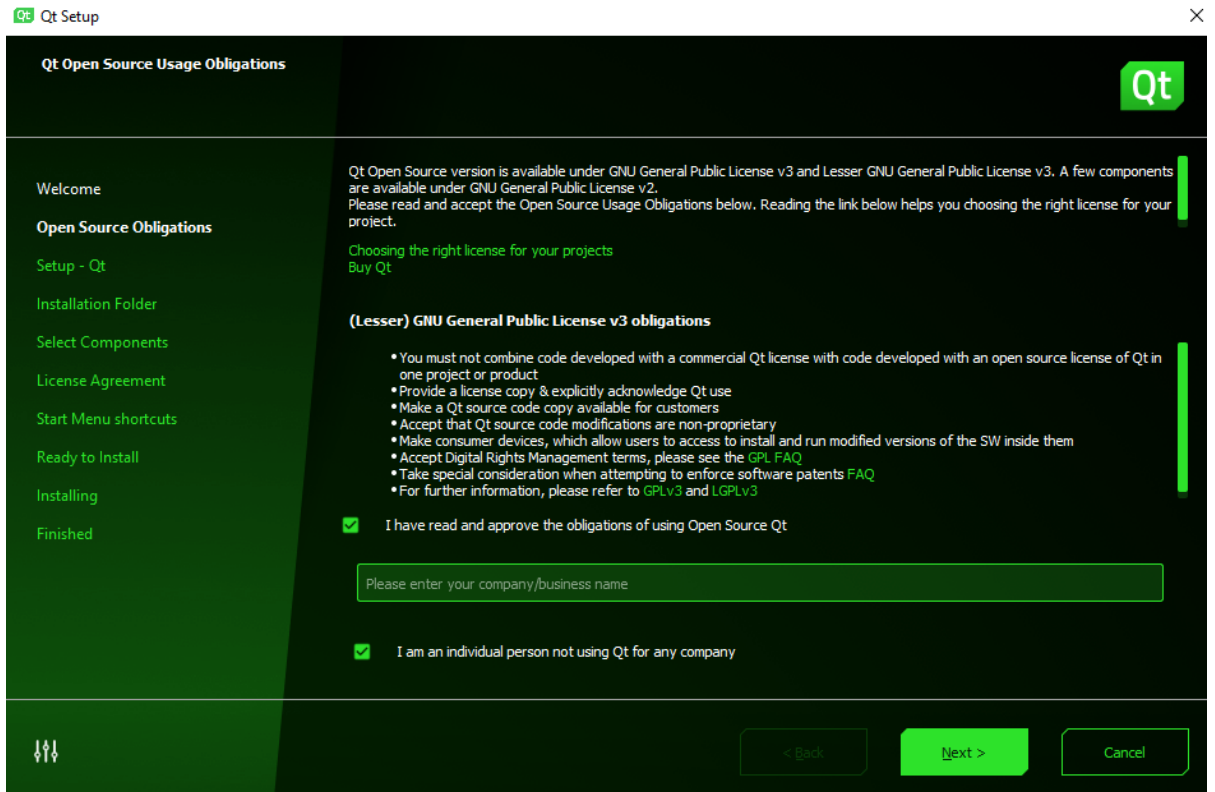
City

Phone

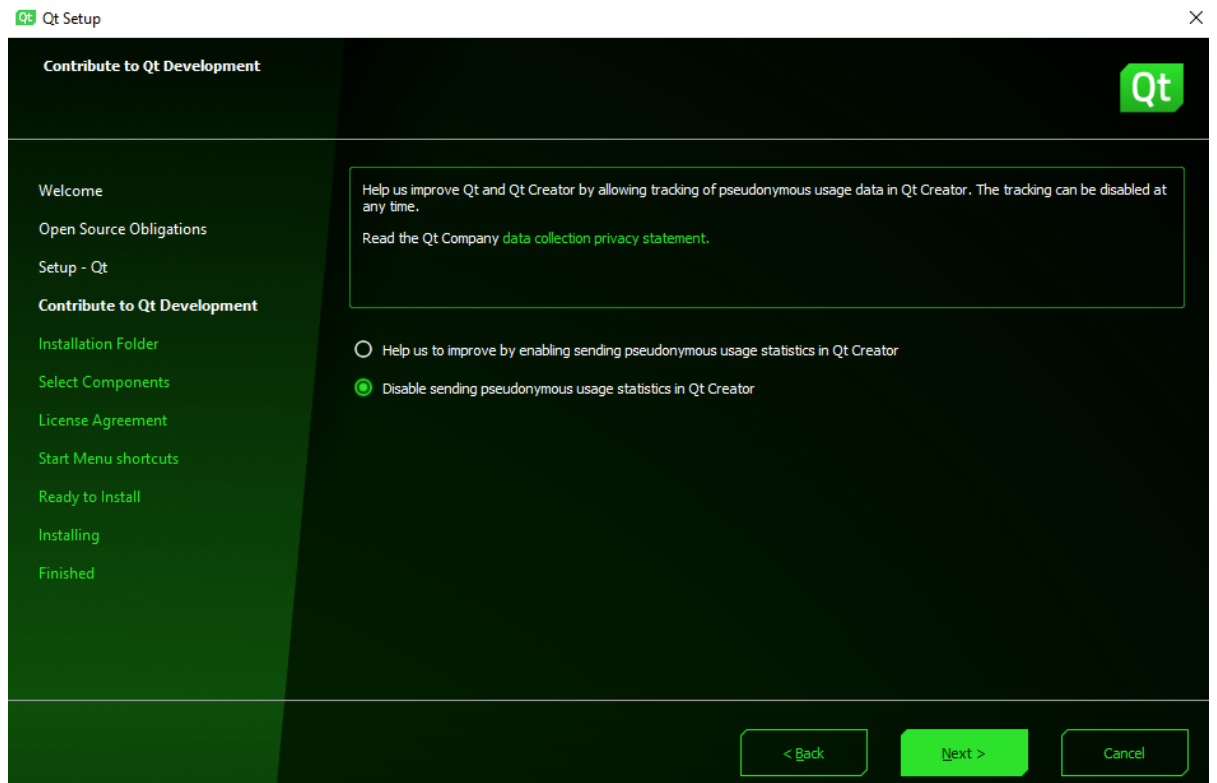
Send me news and information on Qt products and services. Check out our [privacy policy](#).

Confirm

Après le login signer l'agrément de licence



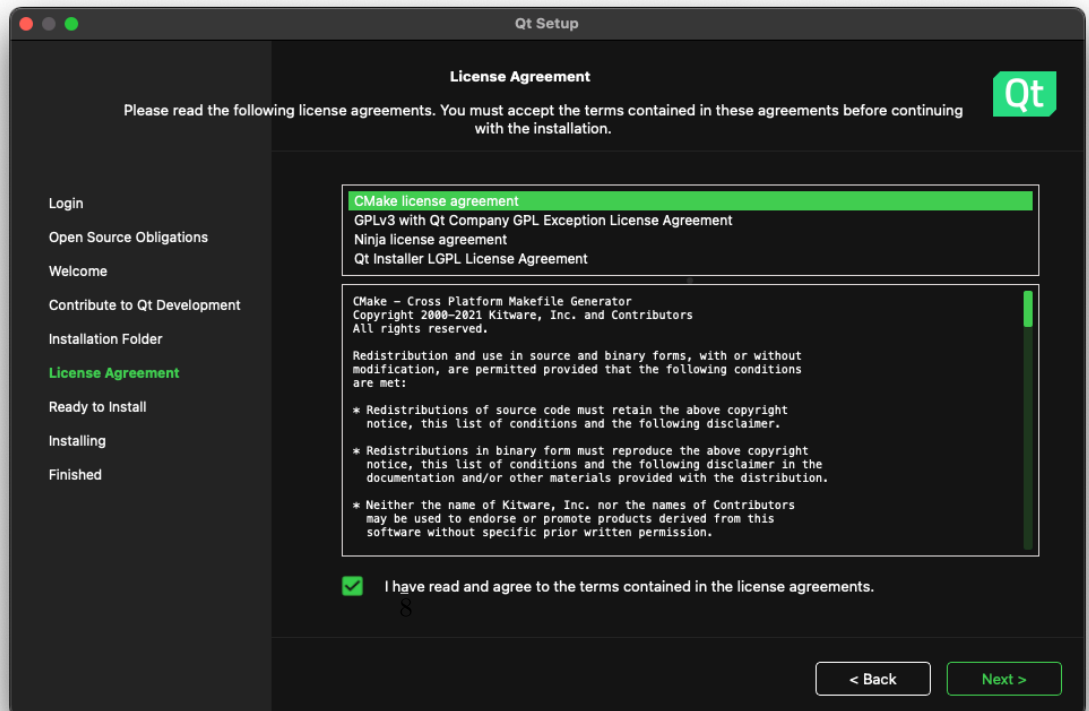
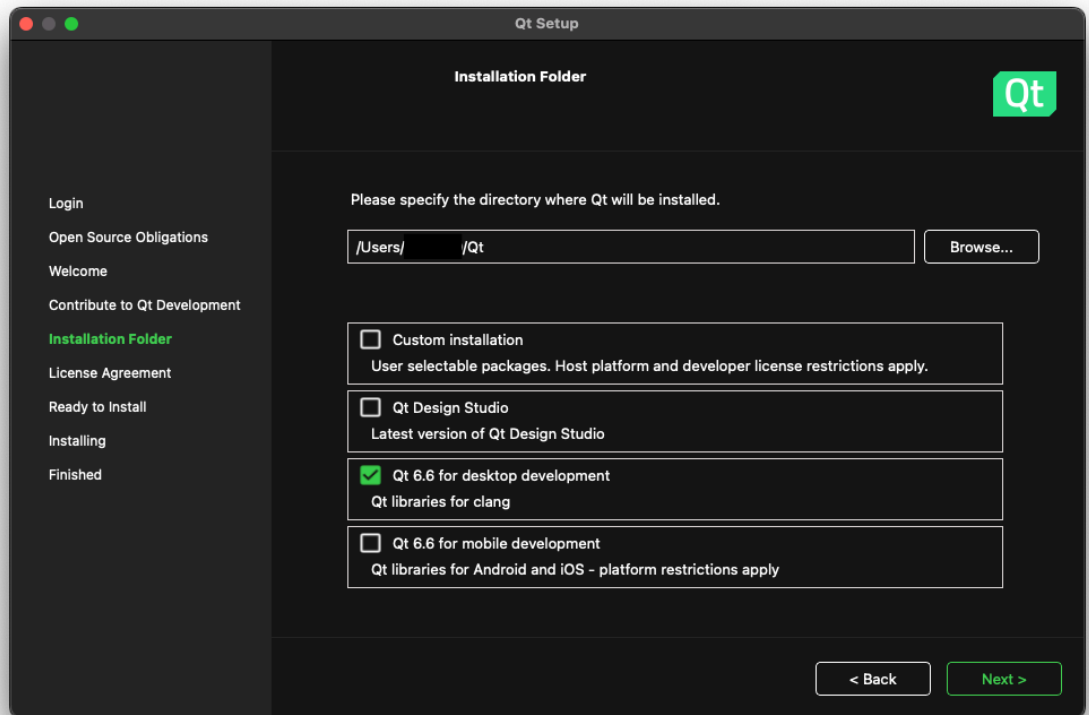
Indiquez votre choix pour l'utilisation des données:



Dans la page “Installation Folder” où vous pouvez définir le chemin d’installation. **Choisissez un chemin simple d’utilisation mais en dehors du dossier de téléchargement.**

**Deux options sont alors possibles** : l’installation “*Desktop development*”, recommandée car plus simple à mettre en place, mais attention lourde en espace (7.46GB), ou l’installation personnalisée \* “Custom installation”\* (voir un peu plus bas).

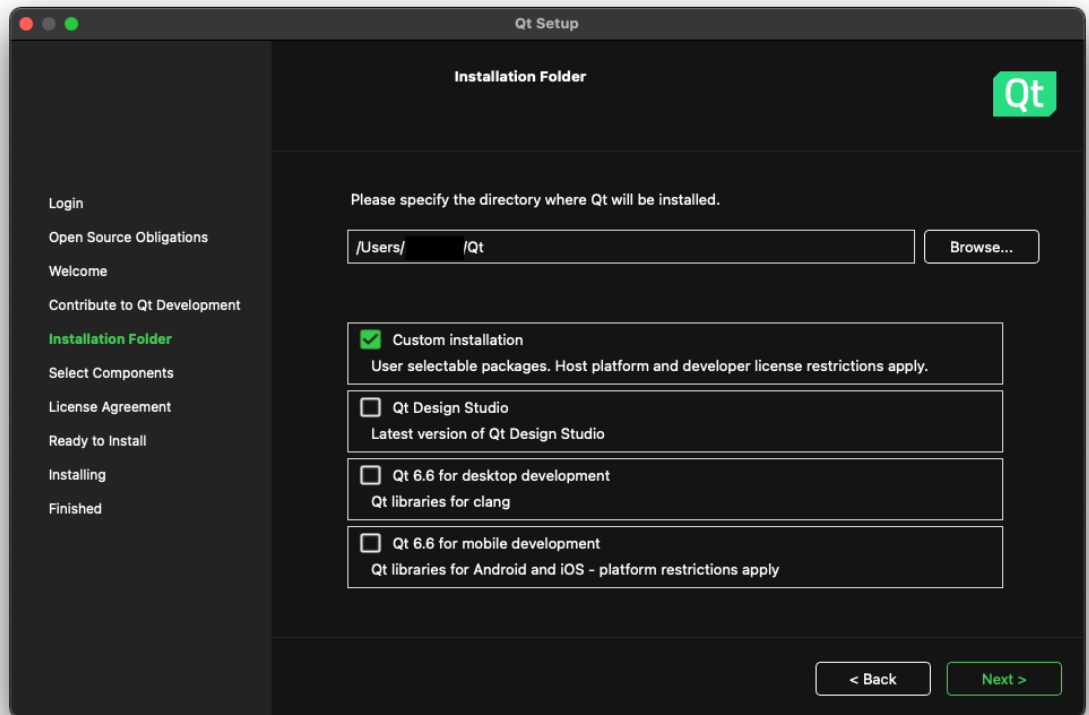
## OPTION 1: Installation “Desktop development”



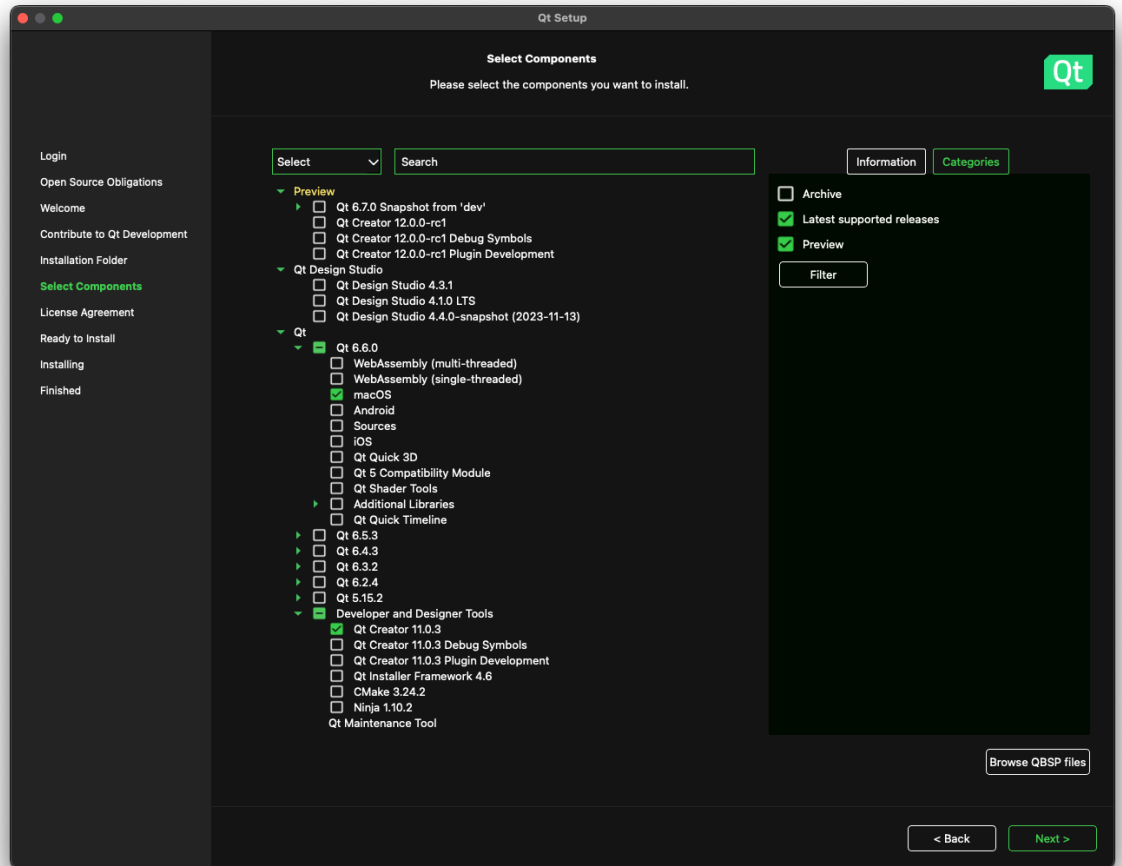


## OPTION 2: Installation personnalisée (alternative à la précédente)

Ce mode d'installation exige moins de mémoire mais est plus complexe à mettre en place.



Sélectionnez les composants requis ( il est important se sélectionner Qt -> Qt 6.6.0 -> macOS et *seulement* Qt creator):



## Problèmes potentiels

Il se peut que vous rencontriez parfois un souci suite à la configuration de votre premier projet. Lorsque vous tentez de sélectionner un kit pendant la création de projet, il peut arriver que cela soit tout simplement impossible et que vous ne parveniez pas à cocher la case du kit en question. QtCreator vous avertira donc : `No suitable kits found`, comme sur l'image ci-dessous.



Si vous êtes face à cette situation, il vous faut cliquer sur le lien bleu “options” et aller dans les paramètres de kit. Une fois dans cette fenêtre, sélectionner le kit en question et QtCreator devrait vous afficher lesdits paramètres juste en dessous de la zone de sélection de kit.

Dès lors vous devez vérifier 4 paramètres dans cette liste :

Les compilateurs C et C++ ("Compiler")

Le debugger ("Debugger")

Votre version de Qt ("Qt version")

Votre outil cmake ("CMake Tool")

Tous les 4 champs doivent posséder une valeur autre que “None” ! Pour ce qui est des valeurs précises que vous devez y trouver, cela dépendra beaucoup de votre OS ainsi que de l’endroit où vous avez installé Qt, cmake, .....

Pour régler votre problème, identifiez le/les champ/s contenant la valeur “None” (impliquant que QtCreator ne sait pas où trouver les outils dont il a besoin) et remplissez les avec les valeurs adéquates.

Dans quasiment tous les cas, les compilateurs seront automatiquement détectés et ne devraient pas poser de problèmes et il en va de même pour le debugger. Cependant, il se peut que vous n’ayez pas installé Qt ou cmake ou bien alors que QtCreator n’a pas réussi à les détecter tout seul sur votre système. Dans ce dernier cas assez pénible, il va vous falloir trouver l’emplacement des exécutables qmake et cmake respectivement. Encore une fois leur emplacement est bien trop dépendant de l’OS pour vous aiguiller ici mais vous pouvez tenter de les chercher sur le Finder de votre Mac.

Une fois les fichiers trouvés, il ne vous reste plus qu'à dire à QtCreator leur emplacement en remplaçant le "None" par le bon path. A partir de là le kit devrait être sélectionnable et utilisable pour la compilation !


## Installation de SFML

L'interface graphique utilisée dans le cadre de ce projet est codée au moyen de la librairie graphique SFML. Pour l'installer, exécutez la commande suivante dans un terminal :

```
brew install SFML
```

## Installation de l'éditeur de texte Geany (optionnel si Qt Creator est choisi comme outil et si Geany est déjà installé)

- télécharger l'installateur Mac OSX à l'adresse <https://www.geany.org/Download/Releases> ;
- lancer l'installation :



Distribution	File	GPG Signature	GPG Key
Source (tar.gz)	<a href="#">geany-1.35.tar.gz</a>	<a href="#">geany-1.35.tar.gz.sig (Instructions)</a>	<a href="#">colombanw-pubkey.txt</a>
Source (tar.bz2)	<a href="#">geany-1.35.tar.bz2</a>	<a href="#">geany-1.35.tar.bz2.sig (Instructions)</a>	<a href="#">colombanw-pubkey.txt</a>
Windows	<a href="#">geany-1.35_setup.exe</a>	<a href="#">geany-1.35_setup.exe.sig (Instructions)</a>	<a href="#">eht16-pubkey.txt</a>
Mac OSX	<a href="#">geany-1.35_osx.dmg</a>	-	-

[Release notes for Geany 1.35](#)