

[Version 1.0]

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

Pour pouvoir programmer sur votre propre machine dans le cadre de ce cours, il vous faudra un environnement nécessaire au développement en C++; à savoir au minimum :

- un compilateur C++ (conseillé : GCC ou Clang) ;
- un éditeur de texte vous permettant de rédiger du code C++ et de l'exécuter (conseillé: Geany).

Cet outillage sera complété par des outils plus avancés lorsque nous aborderons le codage de projets :

- un environnement de développement intégré (EDI, conseillé: Qt Creator) qui remplacera Geany et offrira plus de commodités, notamment pour l'utilisation des outils de «debugging» ;
- un outil de construction de projets: l'outil cmake (<https://cmake.org>) sera employé dans ce cours ;

Cette page a pour but de vous guider dans le processus d'installation de ces outils.

Merci de suivre scrupuleusement les instructions ci-dessous.

**Important: pour éviter tout problème, 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.**

### Détermination de l'architecture

Avant de commencer l'installation de l'environnement, il est crucial de déterminer l'architecture de votre processeur. Pour ce faire, ouvrez une invite de commande (bouton **Windows+R**, puis entrez `cmd`) puis entrez-y la commande **SET Processor**. La variable `PROCESSOR_ARCHITECTURE` devrait contenir **64** dans sa valeur si c'est une architecture 64bits, ou **86** s'il s'agit d'une architecture 32bits. *Souvenez-vous de cette information et restez consistants dans vos choix de versions d'installation* (TOUJOURS 64bit ou TOUJOURS 32bit)

### Créer un dossier d'installation

Les installations qui suivent génèrent un grand nombre de fichiers. En cas de soucis, les retrouver sur votre ordinateur peut s'avérer pénible si les chemins sont définis par défaut. Pour améliorer cela, créez un dossier **Installation**, dans lequel vous pouvez créer un sous dossier par logiciel

installé (Cmake,sfml,MinGW,QT) qui vont contenir les fichiers d'installation de chaque logiciel. Vous pourrez ainsi plus facilement supprimer de mauvaises installations.

## Installation de MinGW (g++, compilateur)

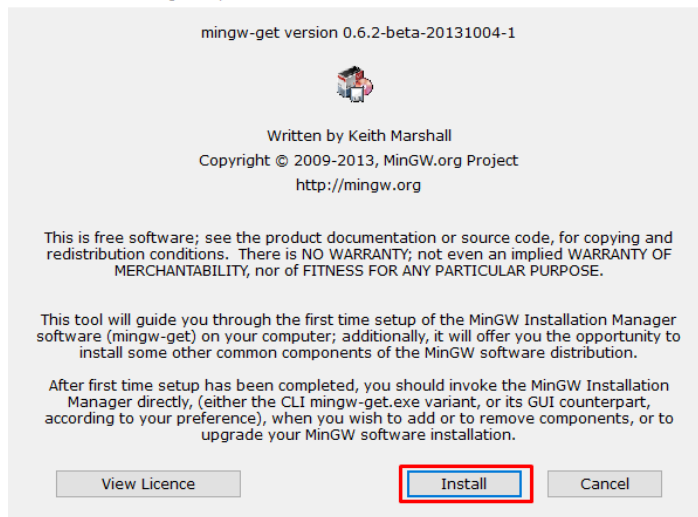
Pour installer le compilateur g++, nous allons installer MinGW :

- Télécharger l'installateur Windows à l'adresse

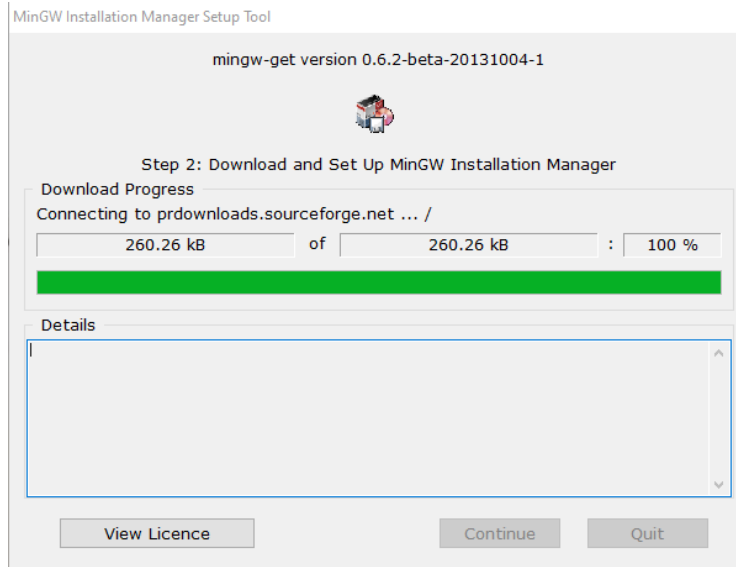
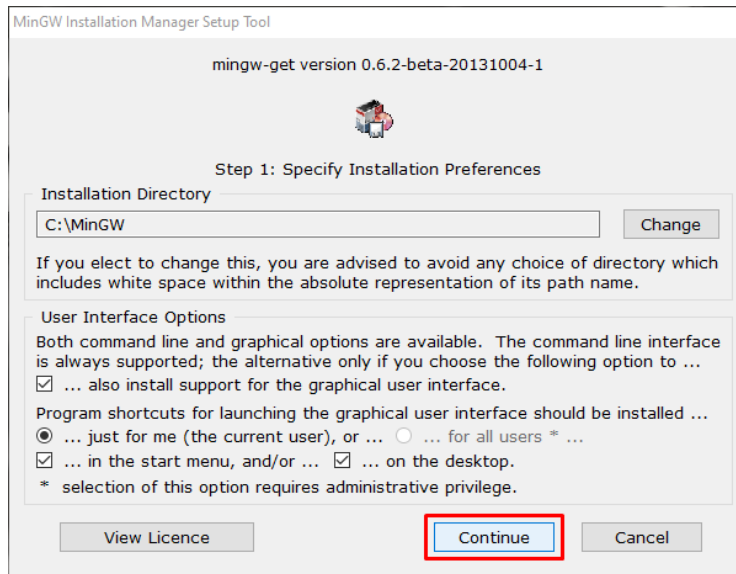
<https://sourceforge.net/projects/mingw/files/latest/download?source=files>

- Lancer l'installation

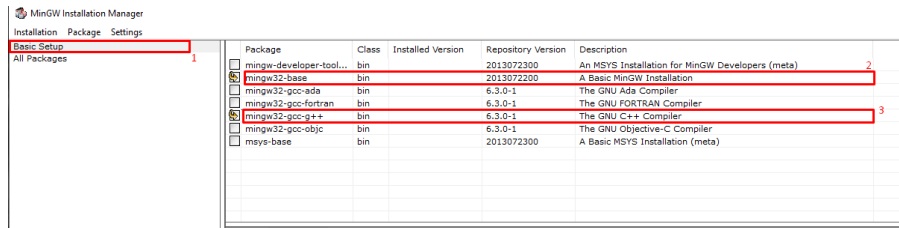
MinGW Installation Manager Setup Tool



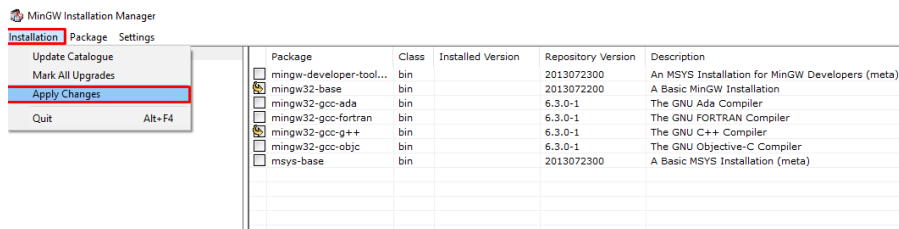
- Remplacer le chemin par défaut C:\MinGW par votre dossier d'installation [...] \Installation\MinGW puis appuyez sur Continue:



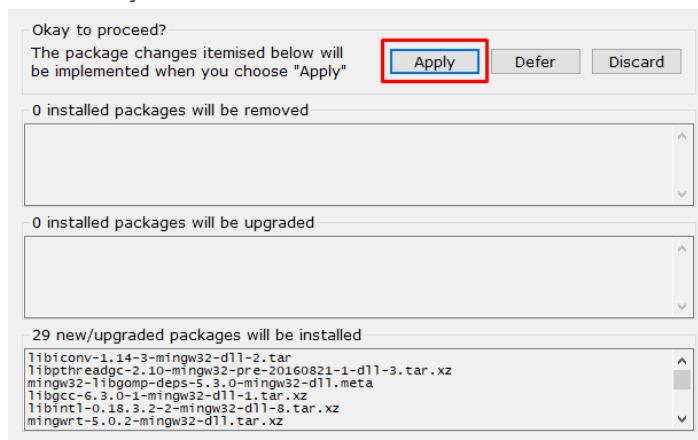
- Dans la fenêtre qui s'ouvre, marquer les packages mingw32-base et mingw32-gcc-g++ (dans la catégorie Basic setup) pour l'installation en cochant la case se situant à gauche puis en cliquant sur **Mark for installation**. Vous pouvez sélectionner ces packages quelque soit votre architecture de processeur (64/32 bits), ceci est la seule exception à la règle.



- Rendez-vous dans le menu Installation > Apply changes, puis cliquez sur Apply pour lancer le téléchargement puis l'installation de g++.



#### Schedule of Pending Actions



Applying Scheduled Changes

All changes were applied successfully; you may now close this dialogue.

☐ Close dialogue automatically, when activity is complete.

Close

Details

```
install: mingwrt-5.0.2-mingw32-dev.tar.xz
installing mingwrt-5.0.2-mingw32-dev.tar.xz
install: binutils-2.28-1-mingw32-bin.tar.xz
installing binutils-2.28-1-mingw32-bin.tar.xz
install: libatomic-6.3.0-1-mingw32-dll-1.tar.xz
installing libatomic-6.3.0-1-mingw32-dll-1.tar.xz
install: libz-1.2.8-1-mingw32-dll-1.tar
installing libz-1.2.8-1-mingw32-dll-1.tar
install: gdb-7.6.1-1-mingw32-bin.tar.lzma
installing gdb-7.6.1-1-mingw32-bin.tar.lzma
install: make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
installing make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
install: gcc-6.3.0-1-mingw32-11c.tar.xz
installing gcc-6.3.0-1-mingw32-11c.tar.xz
install: libstdc++-6.3.0-1-mingw32-dll-6.tar.xz
installing libstdc++-6.3.0-1-mingw32-dll-6.tar.xz
install: gcc-core-6.3.0-1-mingw32-bin.tar.xz
installing gcc-core-6.3.0-1-mingw32-bin.tar.xz
install: mingw32-base-2013072200-mingw32-bin.meta
installing mingw32-base-2013072200-mingw32-bin.meta
install: gcc-c++-4.8.2-mingw32-dev.tar.meta
installing gcc-c++-4.8.2-mingw32-dev.tar.meta
install: gcc-c++-6.3.0-1-mingw32-bin.tar.xz
installing gcc-c++-6.3.0-1-mingw32-bin.tar.xz
```

- Sur Windows 10 ou 11, il faut également installer pthreads pour compiler. Pour ceci il suffit de marquer le paquet mingw32-pthreads-w32, classdev, (dans la catégorie All Packages > MinGW) pour installation, en cochant la case se situant à gauche puis en cliquant sur **Mark for installation**.

MinGW Installation Manager

Installation Package Settings

Package	Class	Installed Version	Repository Version	Description
<input type="checkbox"/> mingw32-pexports	bin		0.47	Generate .def files for DLLs
<input type="checkbox"/> mingw32-pexports	doc		0.47	Generate .def files for DLLs
<input type="checkbox"/> mingw32-pexports	lic		0.47	Generate .def files for DLLs
<input type="checkbox"/> mingw32-pkginfo	bin		0.6.2-beta-20131...	MinGW Package Identification Tool
<input type="checkbox"/> mingw32-post	doc		1.15-1	Command line option parsing library
<input type="checkbox"/> mingw32-post	lic		1.15-1	Command line option parsing library
<input checked="" type="checkbox"/> mingw32-pthreads-w32	dev		2.10-pre-201608...	POSIX threading library for Win32
<input type="checkbox"/> mingw32-pthreads-w32	doc		2.10-pre-201608...	POSIX threading library for Win32
<input type="checkbox"/> mingw32-pthreads-w32	lic		2.10-pre-201608...	POSIX threading library for Win32
<input type="checkbox"/> mingw32-tcl	bin		8.6.0-1	TCL - Tool Command Language
<input type="checkbox"/> mingw32-tcl	dev		8.6.0-1	TCL - Tool Command Language
<input type="checkbox"/> mingw32-tcl	dll		8.6.0-1	TCL - Tool Command Language
<input type="checkbox"/> mingw32-tcl	doc		8.6.0-1	TCL - Tool Command Language

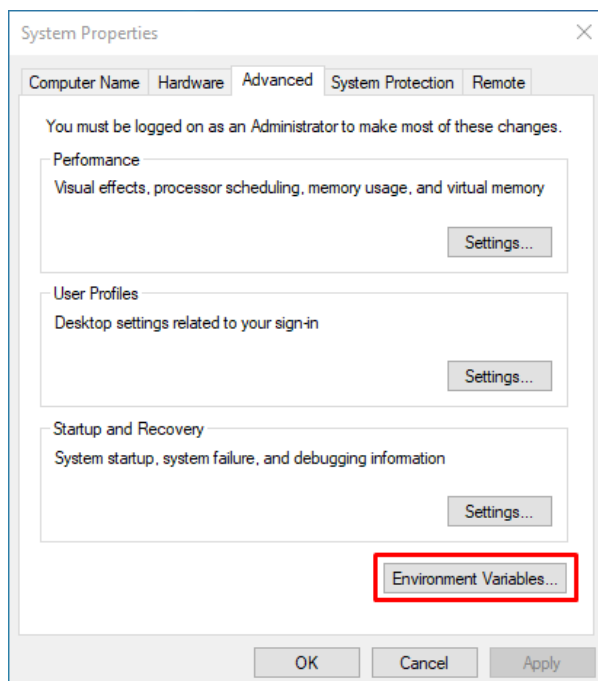
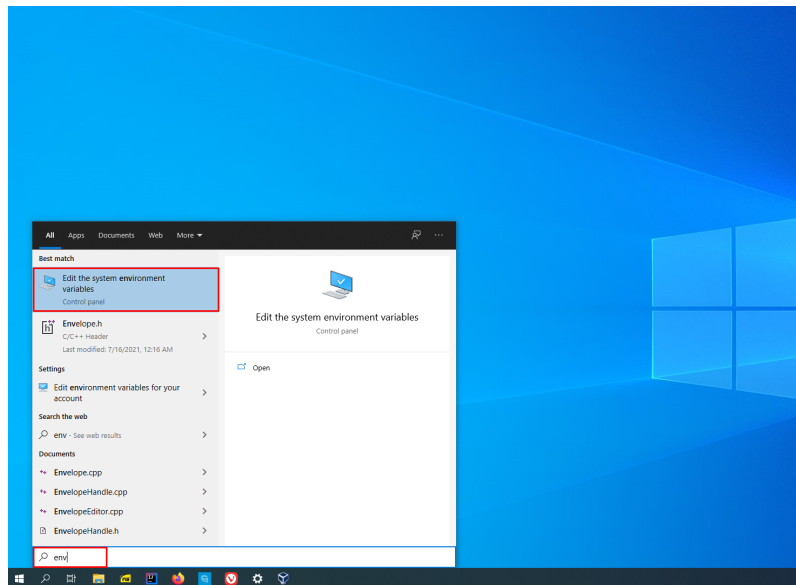
General Description Dependencies Installed Files Versions

**POSIX threading library for Win32**

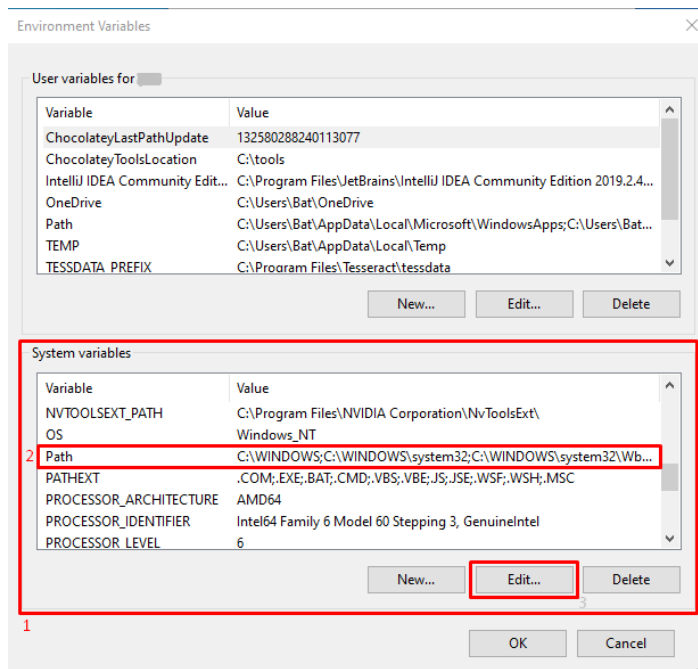
pthread-w32 seeks to provide a freely available and high-quality implementation of pthreads for Windows. Pthreads is an API that is used to create and manage threads in a program. The mingw32-pthreads-w32 package provides the documentation for the pthreads-w32 library, as well as the related development runtime library.

- Rendez-vous ensuite dans le menu Installation > Apply changes, puis cliquez sur Apply.
- Ajouter le répertoire `\bin` du dossier MinGW à la variable d'environnement PATH (voir la vidéo pour les détails). Si vous avez pris la précaution de créer un dossier **Installation**, le chemin à ajouter aura l'allure suivante: `C:\YOUR_PATH\Installation\MinGW\bin`.

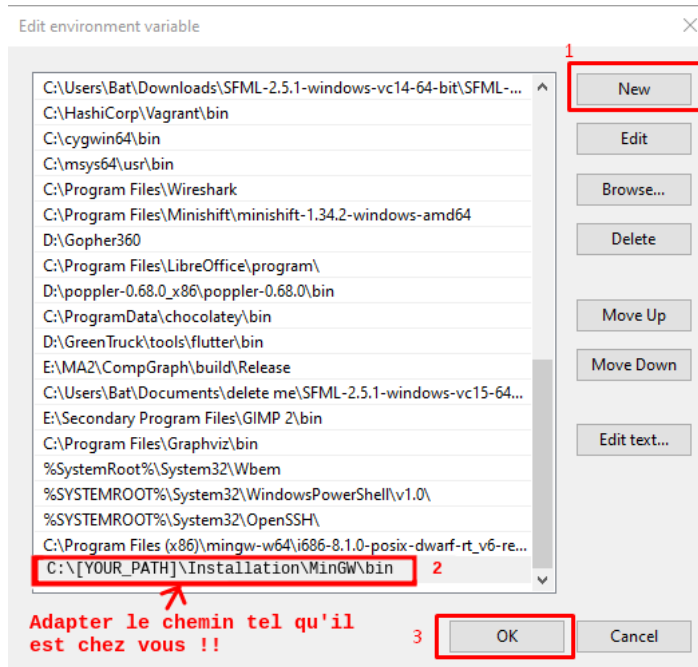
Sur Windows 10 ou 11, cela se fait en recherchant les variables d'environnement :



Editer la variable « Path » :



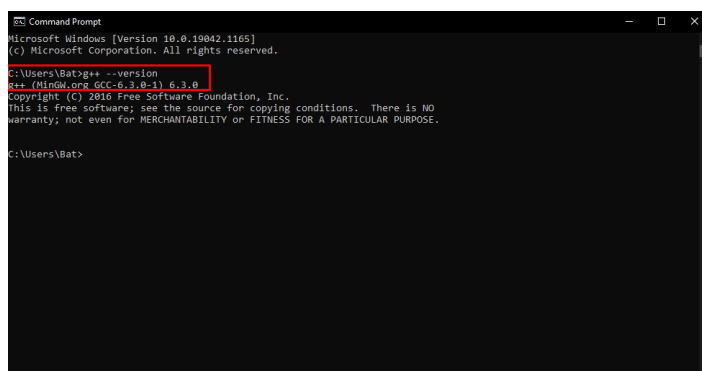
puis ajoutez une nouvelle ligne en bas de la fenêtre :



Vérifiez la version du compilateur en exécutant la commande suivante dans un invite de commandes :

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

Votre version doit être équivalente ou supérieure à 5.4.0.



```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bat>g++ --version
g++ (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (c) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Users\Bat>
```

## Installation de l'éditeur de texte Geany

*Cette installation n'est pas obligatoire si Qt Creator est choisi comme outil de développement et est déjà installé.*

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

Les copies d'écran ci-dessous sont données à titre d'exemple, la version de Geany actuelle est plus récente (2.0), mais l'esprit reste le même.



Distribution	File	GPG Signature
Source (tar.gz)	<a href="#">geany-1.37.1.tar.gz</a>	<a href="#">geany-1.37.1.tar.gz.sig (instructions)</a>
Source (tar.bz2)	<a href="#">geany-1.37.1.tar.bz2</a>	<a href="#">geany-1.37.1.tar.bz2.sig (instructions)</a>
Windows	<a href="#">geany-1.37.1_setup.exe</a>	<a href="#">geany-1.37.1_setup.exe.sig (instructions)</a>
Mac OSX	<a href="#">geany-1.37_osx-3.dmg</a>	-

[Release notes for Geany 1.37.1](#)





## Welcome to Geany 1.37.1 Setup

Setup will guide you through the installation of Geany 1.37.1.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Next >

Cancel

### License Agreement

Please review the license terms before installing Geany 1.37.1.



Press Page Down to see the rest of the agreement.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

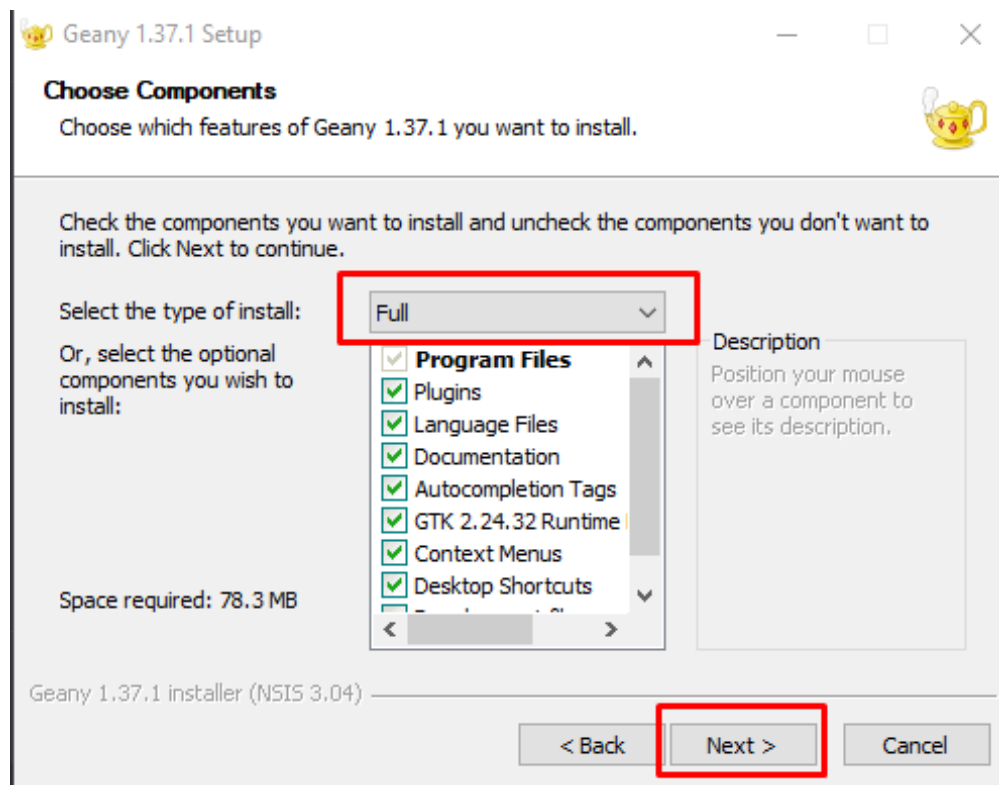
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Geany 1.37.1.


Geany 1.37.1 installer (NSIS 3.04)

< Back

I Agree

Cancel



 Geany 1.37.1 Setup



### Choose Install Location

Choose the folder in which to install Geany 1.37.1.



Setup will install Geany 1.37.1 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files (x86)\Geany

Browse...

Space required: 78.3 MB

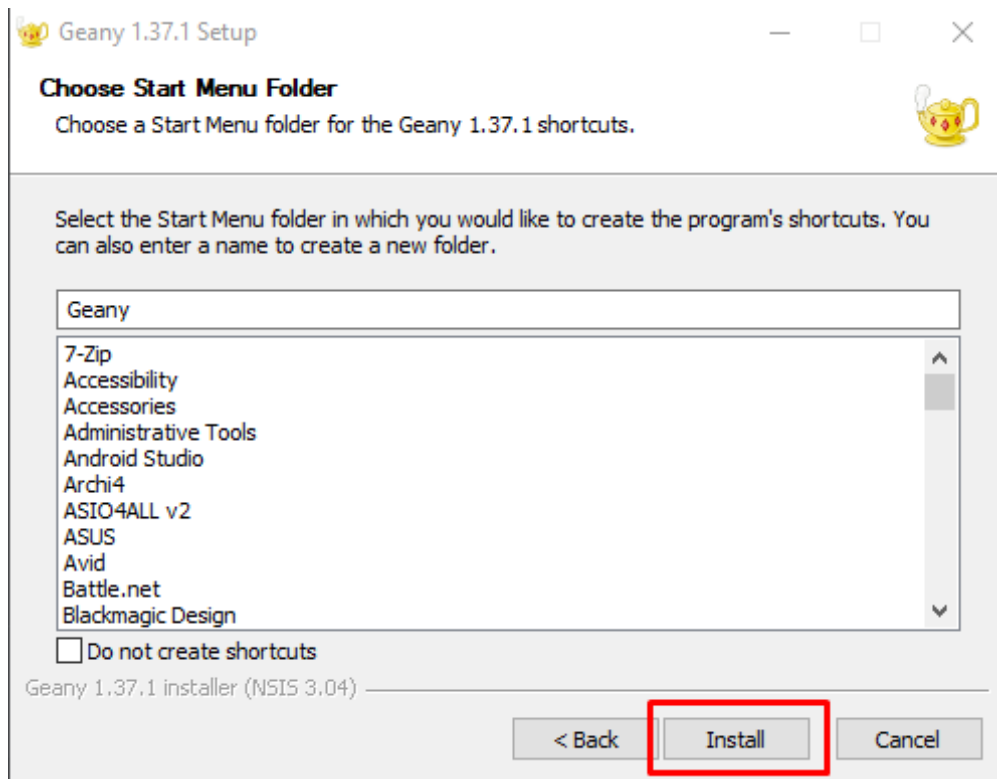
Space available: 21.8 GB

Geany 1.37.1 installer (NSIS 3.04)

< Back

Next >

Cancel



**Note:** pour les systèmes 32-bits il s'agira d'une version plus ancienne. Il est aussi nécessaire d'autoriser Geany à apporter des modifications à l'appareil.

**L'environnement précédemment décrit est suffisant pour faire vos premiers pas en programmation. Il est cependant recommandé de procéder aux installations suivantes lorsque les bases auront été amorcées, et ce, en vue de vous préparer au codage de projets.**

## Installation de CMake

Pour coder le projet de ce cours nous vous fournissons une coquille de base contenant un ensemble de fichiers à utiliser. Cette coquille contiendra en particulier des fichiers de configuration permettant de compiler le projet au moyen d'un utilitaire appelé CMake qu'il vous faut installer.

- Télécharger l'installateur Windows à l'adresse <https://cmake.org/download>.

← → ↻ 🔍 https://cmake.org/download/ CMake About Resources Developer Resources Download 🔍 KITWARE IS HIRING

**Latest Release (3.21.2)**

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with ./bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untarred in any directory. They are prefixed by the version of CMake. For example, the linux-x86\_64 tar file is all under the directory cmake-linux-x86\_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.txt at the top of the source tree. See also the CMake 3.21 Release Notes.

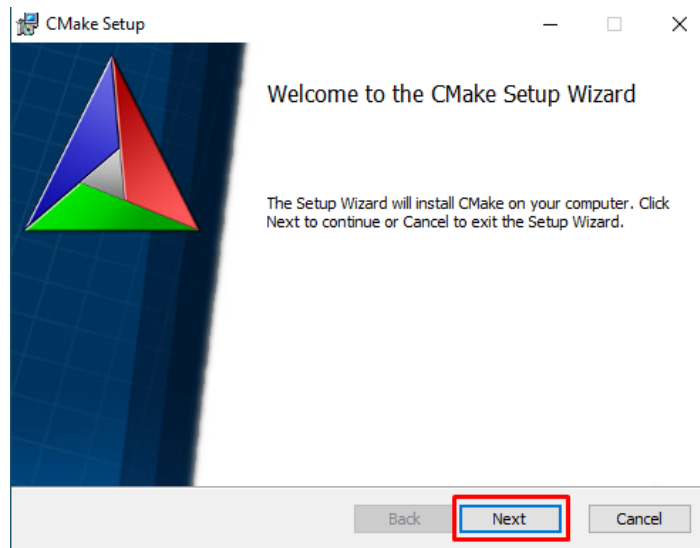
Source distributions:

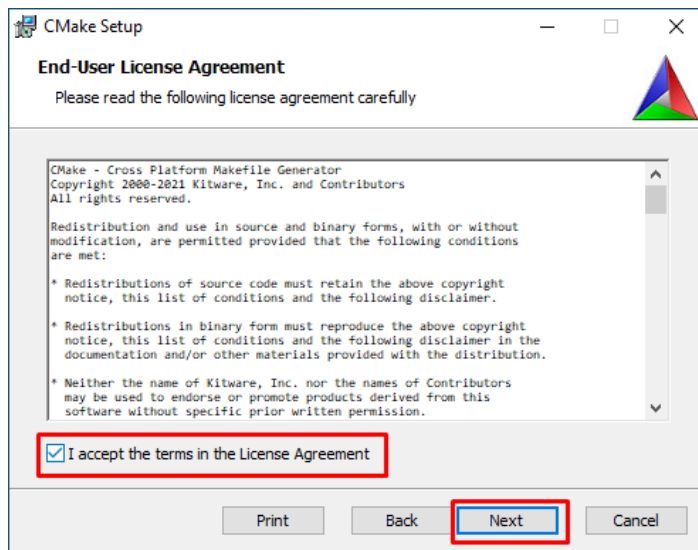
Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.21.2.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.21.2.zip

Binary distributions:

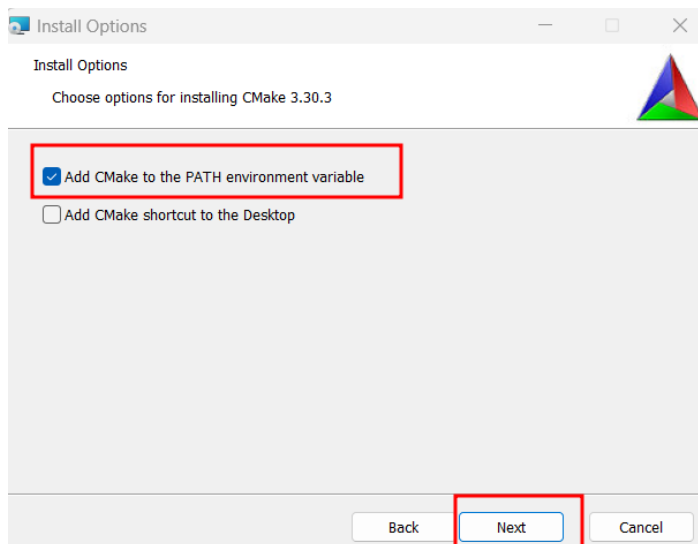
Platform	Files
Windows x64 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	cmake-3.21.2-windows-x86_64.msi <span style="color: red;">64 bits</span>
Windows x64 ZIP	cmake-3.21.2-windows-x86_64.zip
Windows i386 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	cmake-3.21.2-windows-i386.msi <span style="color: red;">32 bits</span>
Windows i386 ZIP	cmake-3.21.2-windows-i386.zip
macOS 10.13 or later	cmake-3.21.2-macos-universal.dmg
	cmake-3.21.2-macos-universal.tar.gz
macOS 10.10 or later	cmake-3.21.2-macos10.10-universal.dmg
	cmake-3.21.2-macos10.10-universal.tar.gz
Linux x86_64	cmake-3.21.2-linux-x86_64.sh

- Lancer l'installation

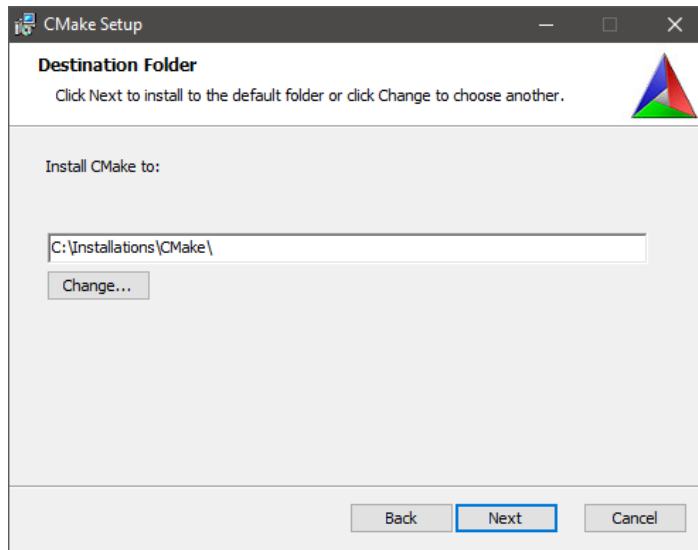




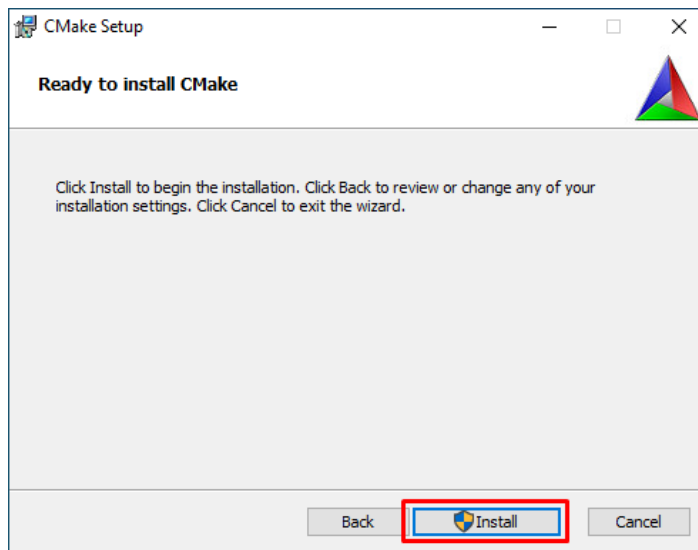
- Ajouter CMake à la variable PATH de Windows



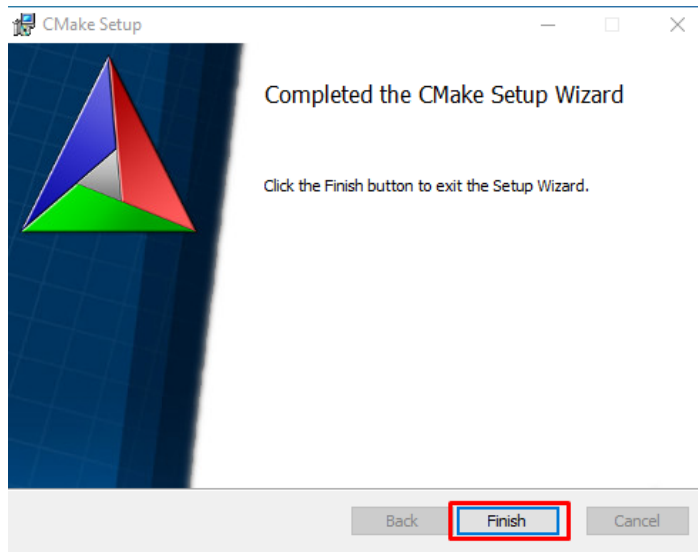
- Modifier le chemin par défaut C:\ProgramFile\CMake par votre dossier d'installation [...] \Installation\CMake, par exemple:



- Installer CMake



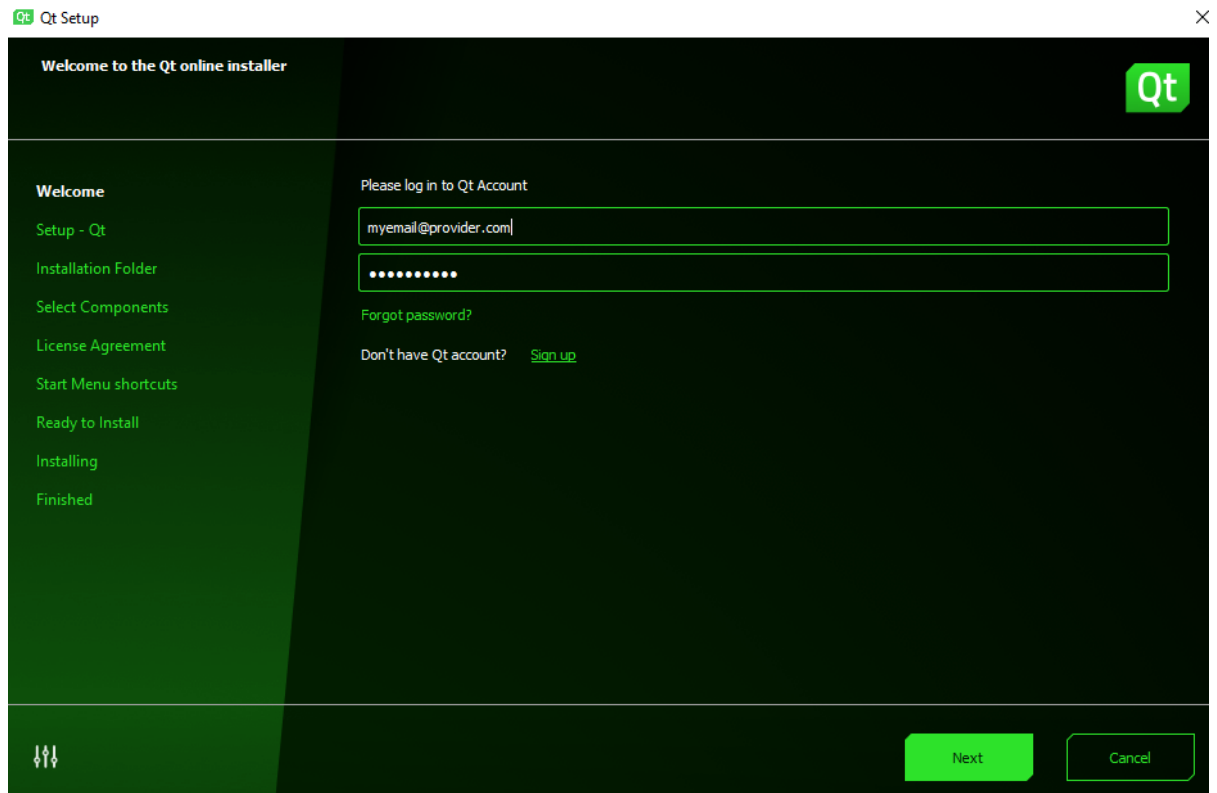




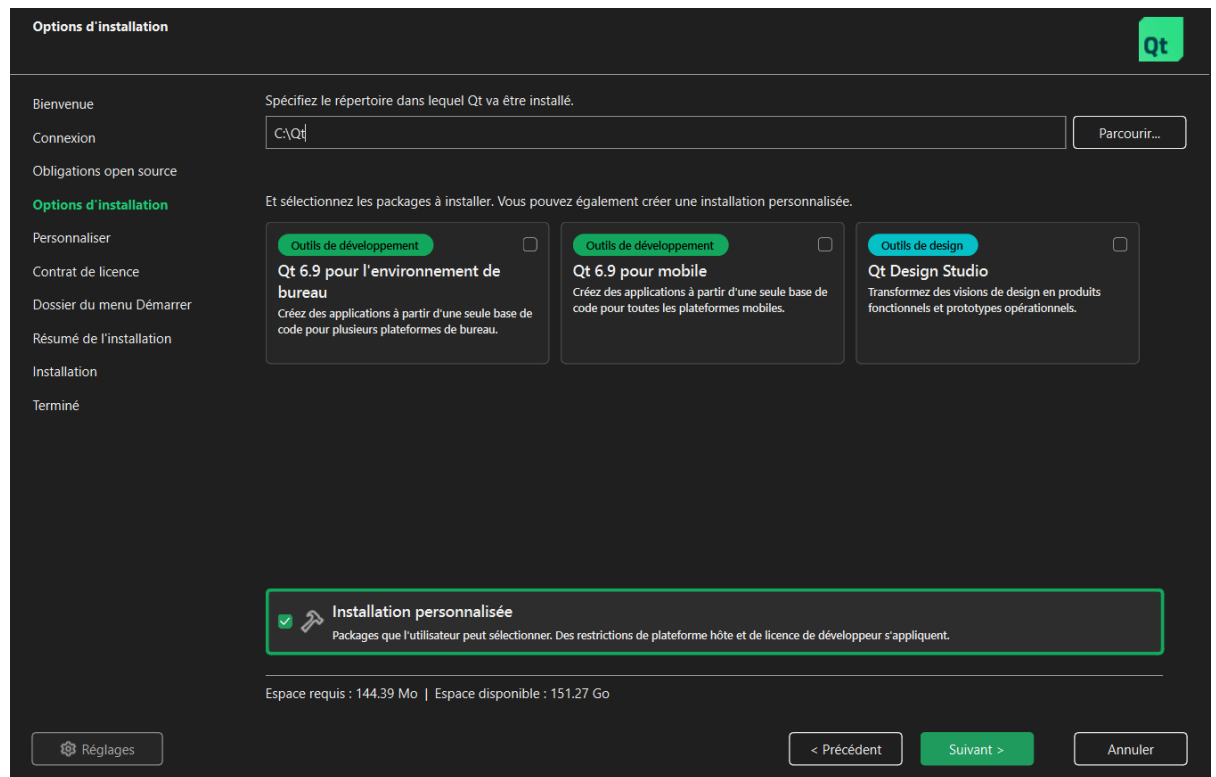
*Attention: Notez le chemin d'installation de Cmake. Lors de la création de projets sur Qt, vous pourriez avoir besoin de spécifier le chemin du répertoire \bin de Cmake si les modules Qt ne sont pas selectionnables (si cela est le cas, sous "Kits" lors de la création de projet, passez votre souris sur le kit grisé, cliquez sur "Manage", puis sous CMake Tool, spécifiez le chemin susmentionné).*

## Installation de Qt Creator

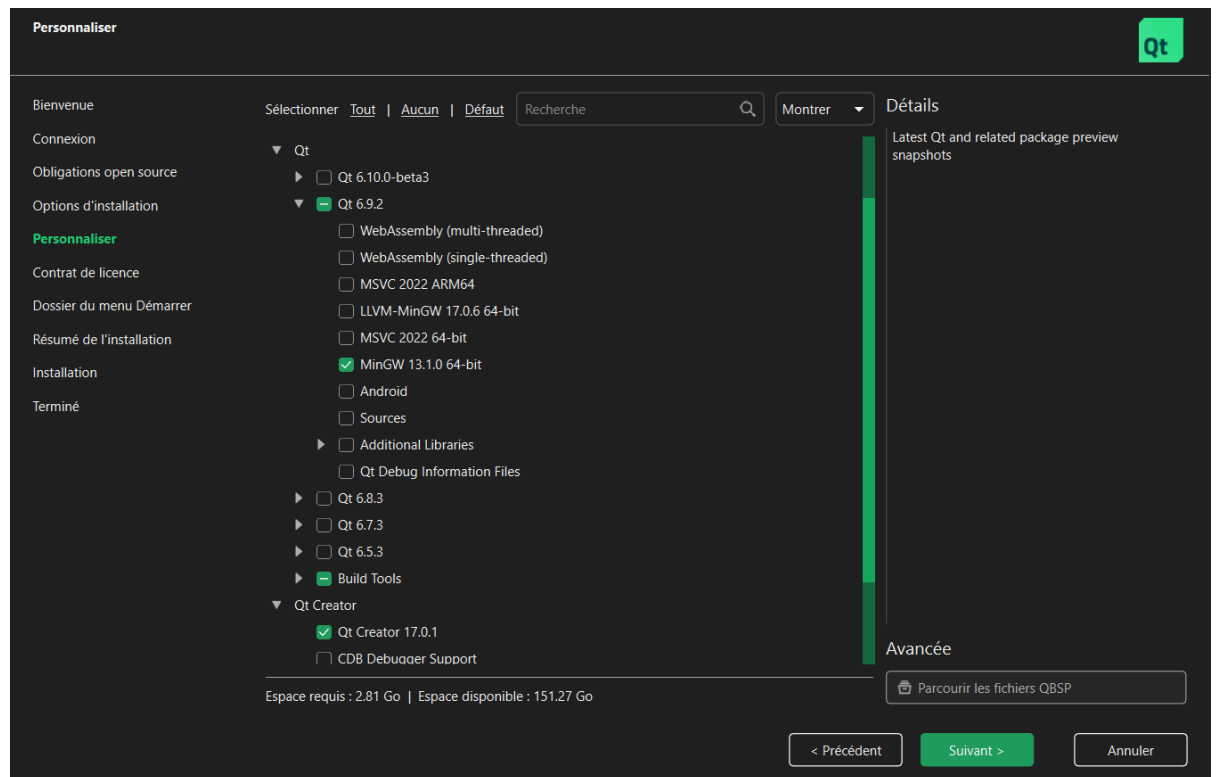
- Télécharger l'installateur Windows à l'adresse <https://www.qt.io/download-ad-qt-installer-oss> ;
- Lancer l'installation Vous aurez besoin de vous authentifier ou de créer un compte pour accéder à l'installateur, utilisez une adresse mail valide puis authentifiez vous via l'installateur.



Dans la page “Installation Folder” où vous pouvez définir le chemin d’installation (choisissez un chemin simple d’utilisation, comme `C:\Qt`), sélectionnez “Custom installation” et désélectionnez “Qt 6.x for desktop environment”.



Lorsque lors de l'installation vous arrivez à un écran "Sélectionner des composants", cliquez sur la petite flèche à côté de la version de Qt que vous choisissez (généralement marquée comme « qt6.xx.x ») puis cochez la case de MinGW qui est sous la version choisie (qt->MinGW). Veillez à bien sélectionner une version de MinGW compatible avec votre architecture! (64/32 bits). Et enfin, en bas de la page, sous qt creator, sélectionnez qt creator 17.xx.x (cf. figure ci-dessous).



Accepter les termes de la licence et cliquer ensuite **Next** plusieurs fois. Enfin autoriser **Qt** à apporter des modifications à l'appareil.

## 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")

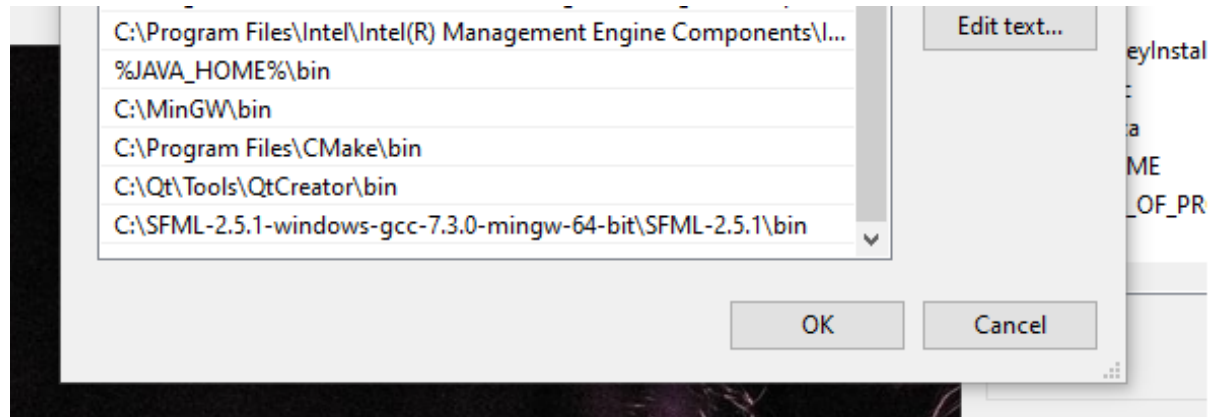
Tous les 4 champs doivent posséder une valeur autre que “None” ! Pour ce qui est des valeurs précises que vous devez y trouver, cela dépendra beaucoup de votre OS ainsi que de l’endroit où vous avez installé Qt, cmake, .....

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 via l’explorateur de fichier sous Windows (chercher qmake.exe et cmake.exe).

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 !

Pour Windows, assurez-vous que les *environment variable PATH* pour MinGW , CMake , et QtCreator soient bien définis.



Le problème est quasiment toujours `Qt\Tools\QtCreator\bin` qui manque.