SYNDICAT QUÉBÉCOIS DE LA CONSTRUCTION SIMULATOR 2023:
Documentation Développeur

Par
Louis-Charles Gaumond
Jonathan Trottier
Claudel D. Roy
Frédéric Léger
Kyle Lussier
Mathieu Duval
Marc-Éric Martel

Travail remis à
Arthur Ouellet
Enseignant

Dans le cadre du cours 420-5DE-HY
Évolution des applications
Groupe 1

Cégep de Saint-Hyacinthe
13 décembre 2023

ENVIRONNEMENT DE DÉVELOPPEMENT

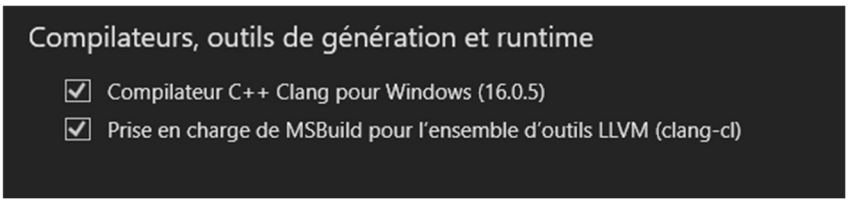
Assurez-vous d'avoir une version à jour de Visual Studio ainsi que les outils nécessaires pour l'utilisation et le développement de logiciel en C++.

Pour cela, ouvrez votre Visual Studio et accédez à l'onglet « Outil » dans le haut de votre application et sélectionnez Outil et fonctionnalités.

Assurez-vous d'avoir une version à jour de Visual Studio 2022 Entreprise (canal « Current »; version minimale 17.8.3) ainsi que les outils nécessaires pour l'utilisation et le développement de logiciel en C++; ceci inclus le Développement Desktop en C++ ainsi que Clang/LLVM (dans les Composants individuels) qui doit être ajouté aux composantes Visual Studio à partir de l'installeur. Il faut aussi avoir Git for Windows ainsi que GitHub Desktop pour la gestion des repositoires sur Windows.

Pour accéder à la solution complète du projet, veuillez cloner le projet à cette adresse : https://github.com/CegepSTH/SQCSim2023.git

Vous pouvez par la suite démarrer la solution. Assurez-vous d'être sur la branche Master.



Les librairies sont déjà dans le repositoire Git pour la version Windows; Pour la version Linux, certaines applications et librairies doivent être préalablement installés au travers du gestionnaire de paquets de la distribution Linux :

1. CMake 3.18.4
2. gcc version 12 à jour.
3. OpenGL (version la plus récente; minimum 1.1.1)

4. SFML 2.5.1 64-bit et 32-bit

5. Devil 1.8.0

6. GLEW 2.1.0

7. irrKlang 1.6.0 version 64-bit et 32-bit

8. Git (dernière version)

Les versions de Windows compatibles sont Windows 10 22H2 et plus récent ainsi que Windows 11 22H2. Les distributions de Linux compatibles sont Ubuntu, Debian, Arch-Linux et Gentoo.

Les paramètres et options de compilation sont gérés par la configuration Visual Studio pour les builds Windows et par CMake pour les builds de Linux.

Clang est utilisé pour la compilation des versions Release sur Windows, MSVC pour les versions Debug sur Windows. GCC est utilisé pour toutes les versions sur Linux.

GDB et Valgrind peuvent être utilisés pour déboguer sur Linux, Visual Studio a tous les outils nécessaires pour le débogage (snapshots de mémoire pour les fuites de mémoire, débogage par point, etc…).

Notez bien que l’absence de driver de carte de son fonctionnel va causer l’échec des builds.