Installation de la SDL 2.0

Par Alexandre Laurent

Date de publication : 20 août 2013

Dernière mise à jour : 29 mars 2024

DÉBUTANT

Apprenez à installer et configurer votre premier projet SDL 2.0.

Commentez

Navigation....3
I - Introduction....3
II - Installation....3
II-A - Installation à partir des binaires....3
II-A-1 - Windows....3
II-A-2 - Linux....3
II-A-3 - Mac OS X....3
II-B - Compilation à partir des sources....3
II-B-1 - Autotools....4
II-B-1-a - Échec de l'étape configure....4
II-B-2 - CMake....4
II-B-2-a - En ligne de commande....4
II-B-2-b - Avec l'interface graphique....5
II-B-3 - Projet Visual C....6
III - Création du projet....6
III-A - Arborescence du projet....6
III-B - Premier code SDL 2....7
III-C - Configuration du projet....8
III-C-1 - GCC (Linux)....8
III-C-2 - Code::Blocks (Windows)....8
III-C-3 - Microsoft Visual Studio (Windows)....9
III-C-4 - Xcode (Mac OS)....10
IV - Projets prêts à utiliser....10
V - Problèmes....11
V-A - Linux....11
V-A-1 - « Failed to load driver : swrast »....11
V-A-2 - « error while loading shared libraries: libSDL2-2.0.so.0: cannot open shared object file: No such file or directory »....11
V-B - Windows....11
V-B-1 - « number of arguments doesn't match prototype »....11
V-B-2 - « previous declaration of 'SDL_main' was here| »....11
V-B-3 - « undefined reference to `WinMain@16' »....12
V-B-4 - Échec de l'étape de configuration avec l'interface graphique de CMake....12
V-B-5 - Erreur 0xc000007b....12
VI - Remerciements....12
Navigation....12

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

Navigation

Sommaire Tutoriel suivant : **vos premières fenêtres**

I - Introduction

La nouvelle version de la **Simple DirectMedia Library** (SDL) est **enfin arrivée**. Celle-ci apporte des évolutions non négligeables mettant au goût du jour la bibliothèque. Il est à nouveau intéressant de se pencher sur celle-ci, notamment car la SDL est une bibliothèque C, facile à utiliser et pratique pour créer des jeux vidéo.

Voyons comment l'installer et créer un premier projet SDL 2.

II - Installation

II-A - Installation à partir des binaires

Les binaires sont les fichiers compilés de la bibliothèque et peuvent être utilisés tels quels. Il est **conseillé de les utiliser lorsqu'ils sont disponibles**.

II-A-1 - Windows

Deux fichiers ont été mis en place pour Windows. Vous devez prendre celui correspondant à votre compilateur :

- **Microsoft Visual C++** ;
- **MinGW**.

Il suffit d'extraire l'archive et de placer les fichiers dans votre projet, comme expliqué dans la section : « **Arborescence du projet** ». Si vous avez un système d'exploitation et la suite de compilation en 64 bits, prenez les fichiers présents dans le sous-dossier x64 (ou x86-64 pour l'archive pour MinGW). Sinon (ou si vous doutez), prenez les fichiers dans le sous-dossier x86 (ou i686 pour l'archive pour MinGW).

II-A-2 - Linux

Pour installer la SDL, il suffit d'utiliser le gestionnaire de paquets de votre distribution. Afin de développer avec la SDL, les paquets à choisir sont ceux de la SDL et leur version de développement (souvent appelés devel ou dev).

II-A-3 - Mac OS X

Téléchargez le fichier **.dmg** et installez-le.

II-B - Compilation à partir des sources

Ne faites la compilation à partir des sources que si vous n'avez pas la possibilité de faire l'installation des binaires, présentées ci-dessus.

Pour compiler la SDL 2.0 à partir **des sources**, vous avez plusieurs possibilités :

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

- **autotools** ;
- **CMake** ;
- Projet Microsoft Visual C (Windows seulement).

Les deux premiers outils permettent de générer les fichiers pour compiler un projet, et ceux pour la configuration actuelle de votre machine. Les fichiers générés seront donc des Makefile, ou les fichiers pour le compilateur et l'éditeur de développement installés actuellement sur votre ordinateur. Il ne suffira plus qu'à lancer la compilation dans votre éditeur préféré afin d'obtenir les fichiers finaux de la bibliothèque.

La troisième possibilité n'est compatible qu'avec Windows et l'éditeur Microsoft Visual Studio. En effet, la SDL propose des fichiers .sln, fichiers de projet de Visual Studio, préconfigurés et permettant ainsi de compiler la bibliothèque.

II-B-1 - Autotools

Autotools est un ensemble d'outils multiplateformes permettant de générer des Makefile et de configurer le projet suivant votre machine.

Pour compiler la SDL, il suffit d'exécuter les commandes suivantes dans le dossier des sources de la bibliothèque :

```
./configure
make
```

et si vous voulez l'installer dans votre système :

```
make install
```

Autotools est très simple à utiliser et représente la norme pour les systèmes GNU/Linux et autres Unix. Sous Windows, il sera nécessaire d'avoir cygwin pour générer les Makefile et les exécuter. Il est donc conseillé d'utiliser **CMake** pour compiler la SDL sous Windows.

Si vous ne faites pas l'installation, vous pouvez trouver les fichiers .a (bibliothèque statique) dans le dossier build et le fichier .so (bibliothèque dynamique), dans le dossier build/.libs/.

II-B-1-a - Échec de l'étape configure

Le script configure permet de vérifier que votre système est prêt pour compiler la bibliothèque. S'il échoue, il n'y a pas de quoi s'inquiéter. Souvent, cela est dû à une dépendance manquante. Lisez attentivement le message d'erreur afin de trouver quel est le logiciel (ou bibliothèque) manquant. Une fois que vous avez trouvé, installez-le (par exemple, à l'aide de votre gestionnaire de paquets).

II-B-2 - CMake

CMake est un programme multiplateforme, tout comme autotools, mais permettant de générer les fichiers de projets pour les éditeurs de développement intégrés (Code::Blocks, Microsoft Visual Studio…). De plus, CMake dispose d'une interface graphique, le rendant plus simple à utiliser.

II-B-2-a - En ligne de commande

Pour la SDL, il est nécessaire de créer un nouveau dossier (par exemple, build) à côté de celui extrait. Ensuite, il suffit d'exécuter à partir du dossier build :

```
cmake -G "Unix Makefiles" ../dossier_des_sources_de_la_SDL
```


Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

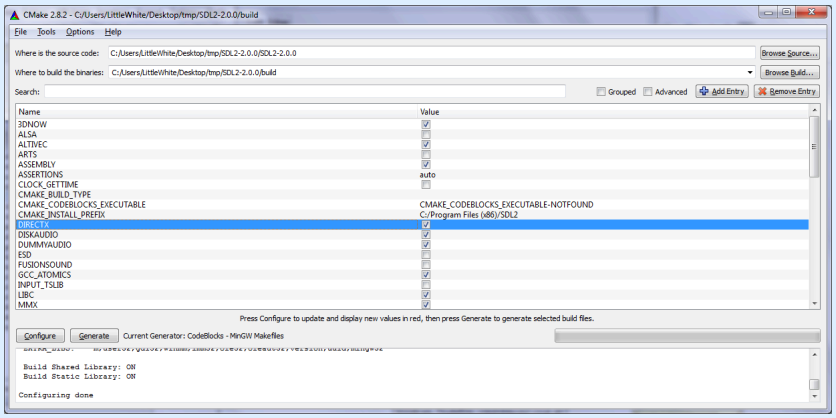
```
make
```

Cet exemple créera un Makefile et est donc parfait pour Linux. L'élément que vous pouvez passer à l'option -G détermine le type de fichier devant être généré. Afin de savoir ce que vous pouvez utiliser comme options, exécutez cmake (sans aucune option) et il vous listera les possibilités.

II-B-2-b - Avec l'interface graphique

Pour MinGW, il faut que le chemin menant au compilateur (dossier bin de MinGW) soit présent dans le PATH. Dans le cas contraire, CMake affichera des erreurs à propos des variables CMAKE_C_COMPILER, CMAKE_C_COMPILER_ENV_VAR…

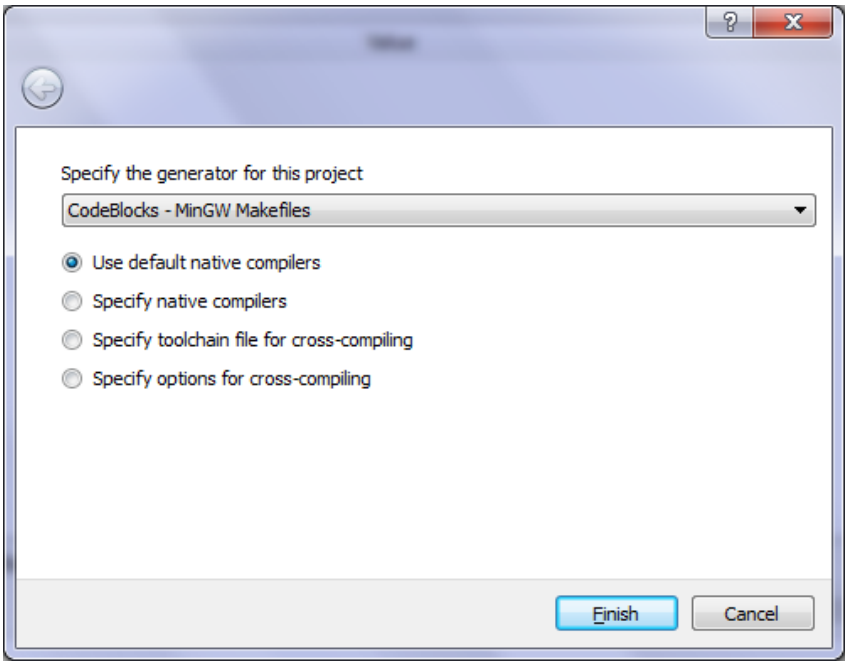
Le **SDK de DirectX** est aussi nécessaire pour la compilation de la SDL sous Windows. Toutefois, vous pouvez vous en passer en décochant la variable DIRECTX.



Une fois l'interface graphique chargée, indiquez le chemin menant au dossier extrait des sources de la SDL 2.0 pour l'entrée « Where is the source code » et indiquez le chemin menant à un dossier build à côté du dossier extrait des sources pour l'entrée « Where to build the binaries ».

Une fois les deux entrées indiquées, vous pouvez cliquer sur le bouton « Configure », choisissez le type de projet à générer et le compilateur à utiliser, puis, si tout s'est correctement déroulé, sur le bouton « Generate ». En cas de réussite, vous aurez les fichiers pour votre éditeur dans le second dossier que vous avez indiqué (« Where to build the binaries »). Pour compiler la bibliothèque, il suffit d'utiliser votre éditeur pour ouvrir et compiler le projet.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.



II-B-3 - Projet Visual C

Les utilisateurs de Microsoft Visual Studio 2008 et supérieures pourront charger le projet de la bibliothèque présent dans le dossier VisualC et ainsi compiler la SDL 2.0. Il suffit d'ouvrir le fichier .sln correspondant à sa version de Visual Studio et de compiler le projet pour obtenir les fichiers .lib et .dll de la bibliothèque.

Si vous obtenez des erreurs indiquant que le compilateur ne trouve pas les fichiers « dinput.h », « dsound.h », vous devez rajouter le répertoire de fichiers d'entête de DirectX à la configuration du projet.

Pour cela, allez dans les propriétés du projet, puis dans le menu C/C++ et rajoutez le répertoire Include du SDL de DirectX (par défaut : C:\Program Files (x86)\Microsoft DirectX SDK (June 2010)\Include\) pour l'entrée « Répertoires d'inclusion supplémentaires » (« Additional Include Directories »).

Afin d'avoir des performances optimales, compilez la bibliothèque en utilisant la cible « Release ». Les fichiers seront ainsi plus légers et optimisés.

III - Création du projet

Maintenant que nous avons les fichiers de la SDL, nous pouvons commencer à créer le projet utilisant la bibliothèque.

III-A - Arborescence du projet

Afin que votre projet soit organisé et facilement distribuable, nous allons créer les dossiers suivants :

- **lib** : pour les fichiers des bibliothèques (.a/.so/.lib) ;
- **include** : pour les fichiers d'entête (notamment, ceux de la SDL) ;
- **src** : pour vos fichiers de code source.

Cette organisation permet d'éviter à ce que vous ayez à installer (ou faire installer) la bibliothèque sur chacun des postes que vous utiliserez : elle est intégrée dans le projet. De plus, l'installation de la bibliothèque dans les

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

répertoires des éditeurs n'est pas recommandée (même si certains sites font de cette façon :(), notamment, car vous forcez l'utilisateur recevant votre projet à installer lui-même la bibliothèque et à reconfigurer le projet, mais aussi, car cela entraîne des oublis. D'autant plus que sous Windows, il n'y a pas d'emplacement déterminé pour mettre les bibliothèques.

Ce qui nous donne l'arborescence suivante :

```
/
include
– SDL2
– – …
– – tous les fichiers .h de la SDL
– – …
lib
– libSDL2-2.0.so (dynamique) ou libSDL2.a (statique) (pour Linux)
– libSDL2.lib (pour Windows) ou libSDL2*.a (pour MinGW)
src
– main.c
SDL2.dll (pour Windows)
les fichiers de projets (.cbp (pour Code::Blocks), .sln (pour Visual Studio)…)
```

Lorsque vous voulez distribuer votre code, il suffit de compresser le dossier et c'est tout. L'utilisateur à qui vous donnez votre projet n'aura qu'à décompresser le fichier et il pourra directement l'utiliser.

III-B - Premier code SDL 2

Afin de tester le bon fonctionnement de votre installation, nous créons une application SDL 2 minimaliste. Le code suivant ouvre une fenêtre pendant trois secondes et quitte :

```
#include <SDL2/SDL.h>

#include <stdio.h>

int main(int argc, char** argv)
{
    /* Initialisation simple */
    if (SDL_Init(SDL_INIT_VIDEO) != 0 )
    {
        fprintf(stdout,"Échec de l'initialisation de la SDL (%s)\n",SDL_GetError());
        return -1;
    }

    {
        /* Création de la fenêtre */
        SDL_Window* pWindow = NULL;
        pWindow = SDL_CreateWindow("Ma première application SDL2",SDL_WINDOWPOS_UNDEFINED,
                                                                  SDL_WINDOWPOS_UNDEFINED,
                                                                  640,
                                                                  480,
                                                                  SDL_WINDOW_SHOWN);

        if( pWindow )
        {
            SDL_Delay(3000); /* Attendre trois secondes, que l'utilisateur voit la fenêtre */

            SDL_DestroyWindow(pWindow);
        }
        else
        {
            fprintf(stderr,"Erreur de création de la fenêtre: %s\n",SDL_GetError());
        }
    }
```


Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

```
    SDL_Quit();

    return 0;
}
```

III-C - Configuration du projet

Une fois que tous les fichiers sont prêts, que le code source est prêt, il ne reste plus qu'à créer le projet de votre nouvelle application SDL 2 afin de la compiler.

Chaque éditeur de développement intégré possède ses propres menus, mais l'idée est toujours la même.

Nous devons spécifier le répertoire contenant les fichiers d'entête (include), le répertoire des fichiers de la bibliothèque (lib) et les bibliothèques à intégrer. Ne pouvant pas décrire ces étapes pour tous les éditeurs existants, seuls les plus connus seront présentés.

III-C-1 - GCC (Linux)

Dans votre ligne de compilation, il suffit de spécifier le répertoire où chercher la bibliothèque (-L./lib), le répertoire des entêtes (-I./include) et d'indiquer que vous voulez utiliser la SDL 2.0 (-lSDL2-2.0).

Ce qui donne :

```
gcc -Wall -Wextra -L./lib -I./include -lSDL2-2.0 src/main.c -o main
```

Lorsque la bibliothèque est installée à l'aide du gestionnaire de paquets, ou de la commande make install, il n'est pas nécessaire de spécifier les dossiers lib et include car les fichiers sont dans les dossiers que le compilateur scanne par défaut.

III-C-2 - Code::Blocks (Windows)

Créez un nouveau projet de type « Application console » (« Console application »).

Nous n'utilisons pas l'assistant de projet (template projet SDL), car celui-ci ne sait que détecter la SDL 1.2 et ne fonctionne donc pas avec la SDL 2.

Ajoutez le fichier main.c du dossier src dans votre projet (le fichier main.c créé par défaut est à retirer du projet et à supprimer).

Allez dans les options de compilation (« Build options »). Vérifiez que vous avez bien sélectionné la globalité du projet dans l'arborescence à gauche et non seulement une des cibles « Debug » ou « Release » afin d'appliquer vos modifications à toutes les cibles. Dans l'onglet « Répertoires de recherche » (« Search directories ») et le sous-onglet « Compilateur » (« Compiler »), ajoutez le répertoire ./include de votre projet afin que le compilateur trouve les fichiers d'entête. Dans le sous-onglet « Éditeur de liens » (« Linker »), ajoutez de la même manière le répertoire ./lib pour que les fichiers de la bibliothèque puissent être trouvés.

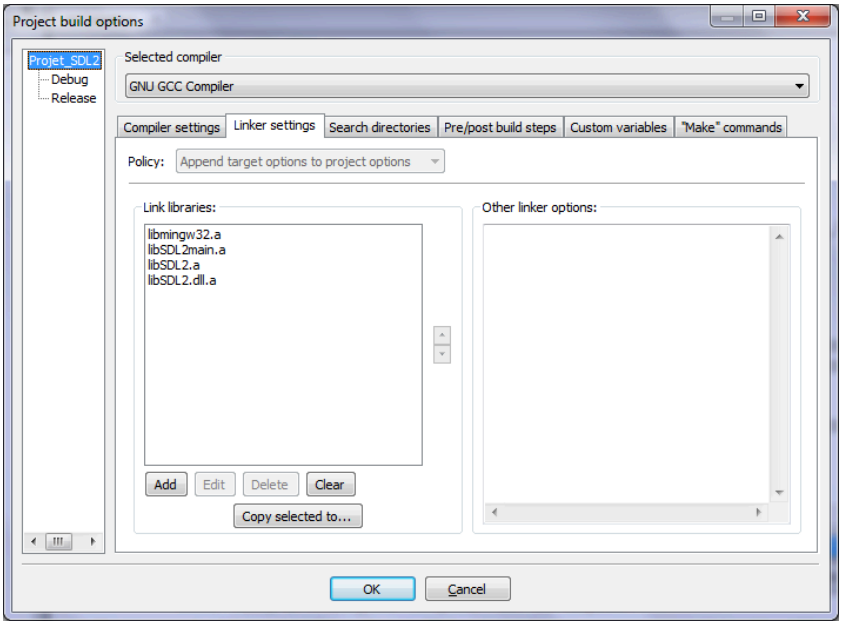
Dans l'onglet « Paramètres de compilation » (« Compiler Settings ») est le sous-onglet « Options de compilation » (« Compileur Flags »), vous pouvez cocher l'option « Activer les avertissements supplémentaires du compilateur [-Wextra] » (« Enable extra compiler warnings [-Wextra] ») afin que le compilateur vous renvoie le maximum d'informations sur votre code et ses possibles erreurs.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

Finalement, dans l'onglet « Paramètres de l'éditeur de liens » (« Linker settings »), vous devez rajouter (dans cet ordre) :

```
libmingw32.a
libSDL2main.a
libSDL2.a
libSDL2.dll.a
```

Comme le montre cette capture d'écran :



Afin que vous puissiez copier votre projet d'un répertoire à un autre et que Code::Blocks continue de trouver les fichiers, il est recommandé d'utiliser des chemins relatifs (par exemple : ./lib et ./include).

Vous devriez maintenant pouvoir compiler le projet.

III-C-3 - Microsoft Visual Studio (Windows)

Créez un nouveau projet « Application Console Win32 » (« Win32 Console Application »). Dans le gestionnaire, indiquez que vous voulez un projet vide (« Empty project »).

Ajoutez le fichier main.c dans le projet.

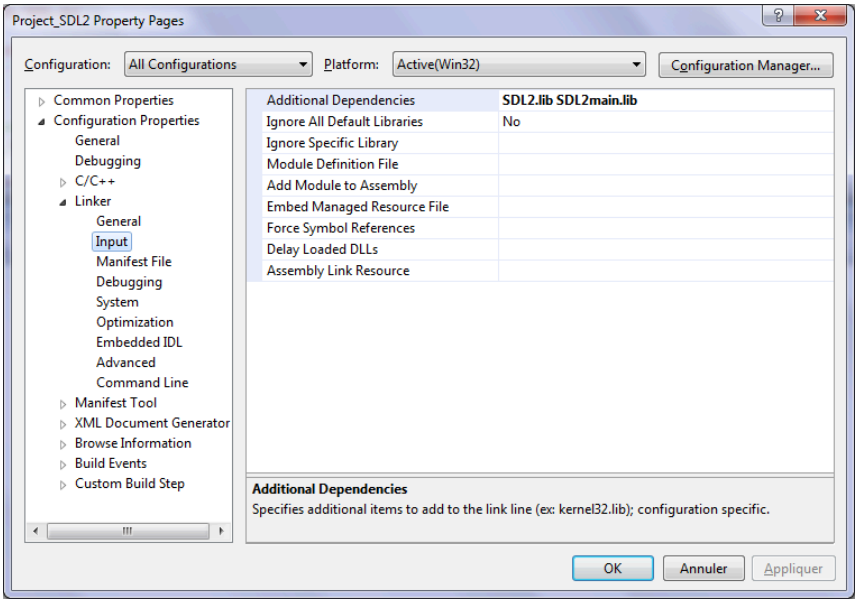
Dans les propriétés du projet « Projet » → « Propriétés Projet SDL 2 » (« Project » → « Projet SDL 2 Properties »), cliquez sur « Propriétés de la configuration » (« Configuration Properties ») afin de déplier l'arborescence et de déverrouiller les deux options en haut de la fenêtre. Sélectionnez la configuration « Toutes plateformes » (« All Configurations ») afin que vos modifications s'appliquent sur la cible « Debug » et la cible « Release ».

Dans le menu C/C++, il est conseillé de changer l'entrée « Niveau d'avertissement » (« Warning Level ») du niveau 3 au niveau 4 afin que le compilateur vous renvoie le maximum d'informations sur votre code et ses possibles erreurs.

Dans le menu C/C++, rajoutez ./include pour l'entrée « Répertoires d'inclusion supplémentaires » (« Additional Include Directories »). Dans le menu « Éditeur de liens » (« Linker »), ajoutez ./lib pour l'entrée « Répertoires de bibliothèque

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

supplémentaires » (« Additional Library Directories »). Dans le sous-menu « Entrée » (« Input »), pour l'entrée « Dépendances supplémentaires » (« Additional Dependencies »), insérez « SDL2.lib SDL2main.lib » :



Pour valider chaque modification des propriétés, cliquez sur le bouton « Appliquer » (« Apply »).

Afin que vous puissiez copier votre projet d'un répertoire à un autre et que Visual Studio continue de trouver les fichiers, il est recommandé d'utiliser des chemins relatifs (par exemple : ./lib et ./include).

Vous devriez maintenant pouvoir compiler le projet.

III-C-4 - Xcode (Mac OS)

Créez un projet « outil en ligne de commande » (« Command Line Tool »). Cliquez sur votre projet dans l'arborescence se trouvant à gauche de l'éditeur afin de faire apparaître les propriétés. Ensuite cliquez sur « Paramètres de compilation » (« Build Settings ») et rajoutez /Libraries/Frameworks à l'entrée « Chemin de recherche des frameworks » (« Framework Search Paths ») permettant ainsi à Xcode de trouver le fichier Framework de la SDL. Dans l'onglet « Étapes de compilation » (« Build Phases ») et dans la sous-section « Lier binaire avec des bibliothèques » (« Link Binary With Libraries »), ajoutez le « Cocoa.framework », puis cliquez sur le bouton « Ajouter autre … » (« Add Other… ») puis dans le dossier /Library/Frameworks, trouvez SDL.framework. Votre projet est maintenant prêt à être compilé.

IV - Projets prêts à utiliser

Afin que vous puissiez démarrer plus rapidement votre projet, voici des archives avec des projets préconfigurés pour différents éditeurs :

- **SDL 2.0.4 - Code::Blocks (MinGW) (32 bits)** ;
- **SDL 2.0.12 – Windows – Microsoft Visual Studio 2019** ;
- **SDL 2.0 - Linux – Makefile (64 bits)** ;
- **SDL 2.0 - MacOS – XCode**.

Ou encore, **ce projet Code::Blocks** avec les extensions SDL Image et SDL TTF.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

Pour mettre à jour pour une version plus récente de la SDL 2, il suffit de télécharger la nouvelle version en ligne et de remplacer les fichiers de la bibliothèque.

Si vous voulez créer une application Windows 64 bits, vous pouvez remplacer les fichiers du dossier lib par leur équivalent 64 bits ainsi que le fichier SDL.dll.

Pour Linux, il est conseillé de recompiler les fichiers du dossier lib ou d'utiliser ceux installés par le gestionnaire de paquets.

V - Problèmes

V-A - Linux

V-A-1 - « Failed to load driver : swrast »

Ce message arrive lorsqu'il n'est pas possible à l'application de charger le pilote graphique. Il est conseillé de réinstaller les pilotes graphiques.

V-A-2 - « error while loading shared libraries: libSDL2-2.0.so.0: cannot open shared object file: No such file or directory »

Si vous n'avez pas installé la SDL avec make install, il suffit d'indiquer explicitement le chemin vers le .so de la SDL en lançant votre application :

```
LD_LIBRARY_PATH=./lib ./main
```

V-B - Windows

V-B-1 - « number of arguments doesn't match prototype »

Avec la version MinGW de la SDL, il est obligatoire de déclarer son main de la forme :

```
int main(int argc, char** argv)
```

L'erreur apparaît si votre main est de la forme suivante :

```
int main()
```

V-B-2 - « previous declaration of 'SDL_main' was here| »

Avec la version MinGW de la SDL, il est obligatoire de déclarer son main de la forme :

```
int main(int argc, char** argv)
```

L'erreur apparaît si votre main est de la forme suivante :

```
int main (void)
```


Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

V-B-3 - « undefined reference to `WinMain@16' »

Pour compiler un programme avec MinGW, il est nécessaire de spécifier la bibliothèque « libmingw32.a » dans l'éditeur de liens, sinon l'erreur « undefined reference to 'WinMain@16' » apparaît.

V-B-4 - Échec de l'étape de configuration avec l'interface graphique de CMake

L'étape de configuration du projet peut échouer lorsque le dossier des sources possède des caractères accentués. Dans un tel cas, il suffit de recommencer la procédure après déplacement du dossier des sources de la SDL.

V-B-5 - Erreur 0xc000007b

Vous êtes sous Windows et votre projet compile correctement. Il se lance, mais avant de faire quoi que ce soit, il vous affiche l'erreur ayant pour obscur code `0xc000007b`. Cela arrive dans le cas d'une incompatibilité entre la DLL que vous utilisez et l'architecture pour laquelle votre programme a été compilé. En effet, SDL 2 propose des fichiers en 32 et en 64 bits. Si vous tentez de faire fonctionner le programme avec la bibliothèque en 64 bits alors que vous utilisez un compilateur 32 bits (comme cela est le cas avec CodeBlocks (MinGW), par défaut) produisant des programmes en 32 bits, alors vous obtiendrez l'erreur. Pour la corriger, il suffit de remplacer les fichiers de la bibliothèque (.DLL, .lib…) par les fichiers en 32 bits et tout rentrera dans l'ordre.

VI - Remerciements

Merci à **Winjerome** pour ses précieuses suggestions ainsi que **ClaudeLELOUP** pour sa correction orthographique.

Navigation

Sommaire

Tutoriel suivant : **vos premières fenêtres**

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2013 Alexandre Laurent. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.