TÉLÉCHARGEMENT ET COMPILATION D'UN PROJET SDL

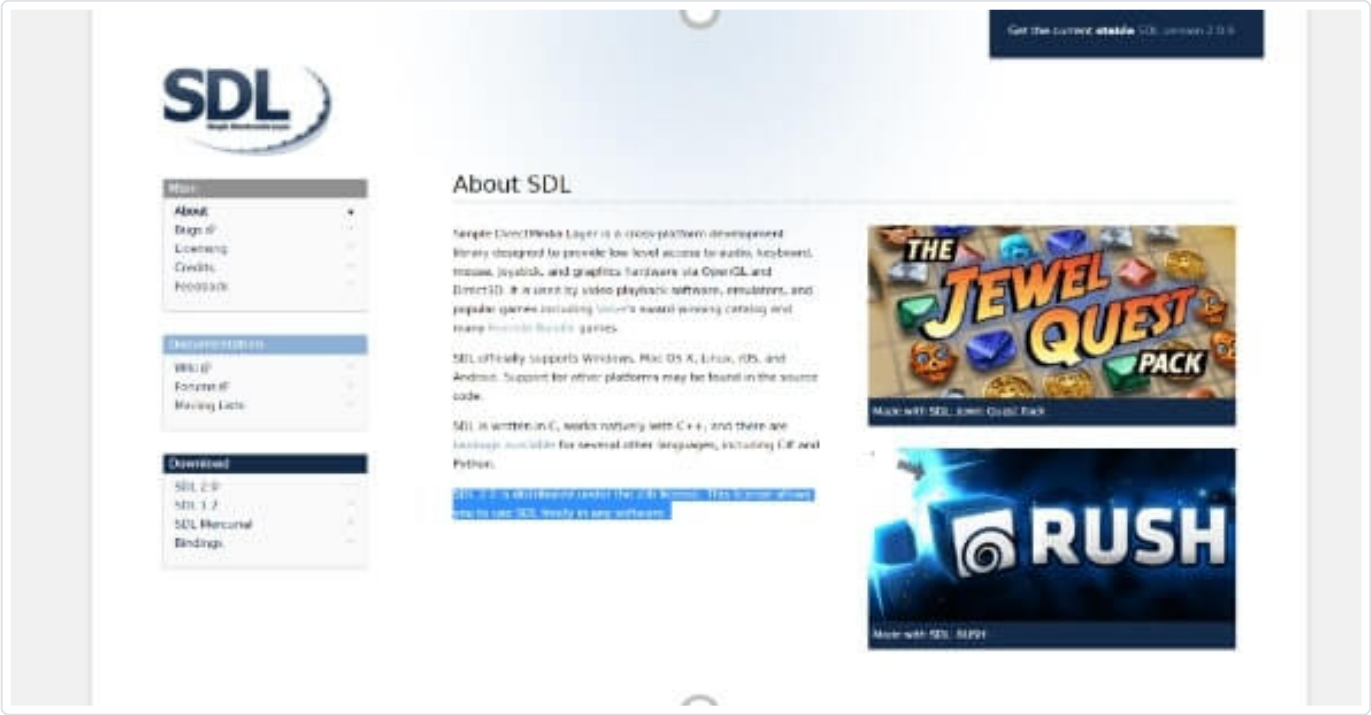
Introduction

Dans cette partie du cours, il va vous falloir vous munir de vos outils de développeur, à savoir : votre **éditeur de code** et votre **compilateur**.

> **Information**
>
> Je suppose que si vous utilisez un IDE alors vous saurez compiler une bibliothèque avec ! Car si vous utilisez un IDE, c'est que vous savez vous en servir...

Téléchargement

Pour commencer on va visiter le site web officiel de la SDL https://www.libsdl.org/, qui ressemble à ceci :

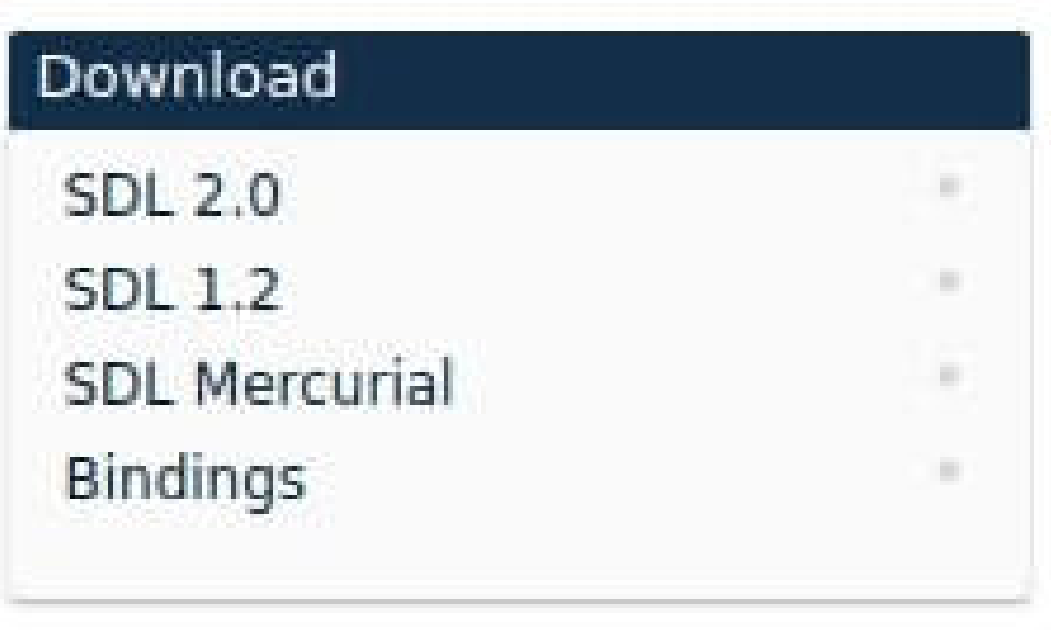


Vous noterez qu'à droite de cette image, sont disponibles les jeux utilisant la SDL (pour les plus curieux d'entre vous lors de chaque visite de la page le catalogue de jeu change). Sachez toutefois que si certains jeux sont en 3D, c'est qu'ils utilisent forcément une API de rendu 3D comme OpenGL, Vulkan, Direct3D, etc...

Au milieu du site web vous retrouverez une courte description en anglais de la bibliothèque.

En haut à droite vous avez la dernière version de la SDL mise en ligne par les développeurs.

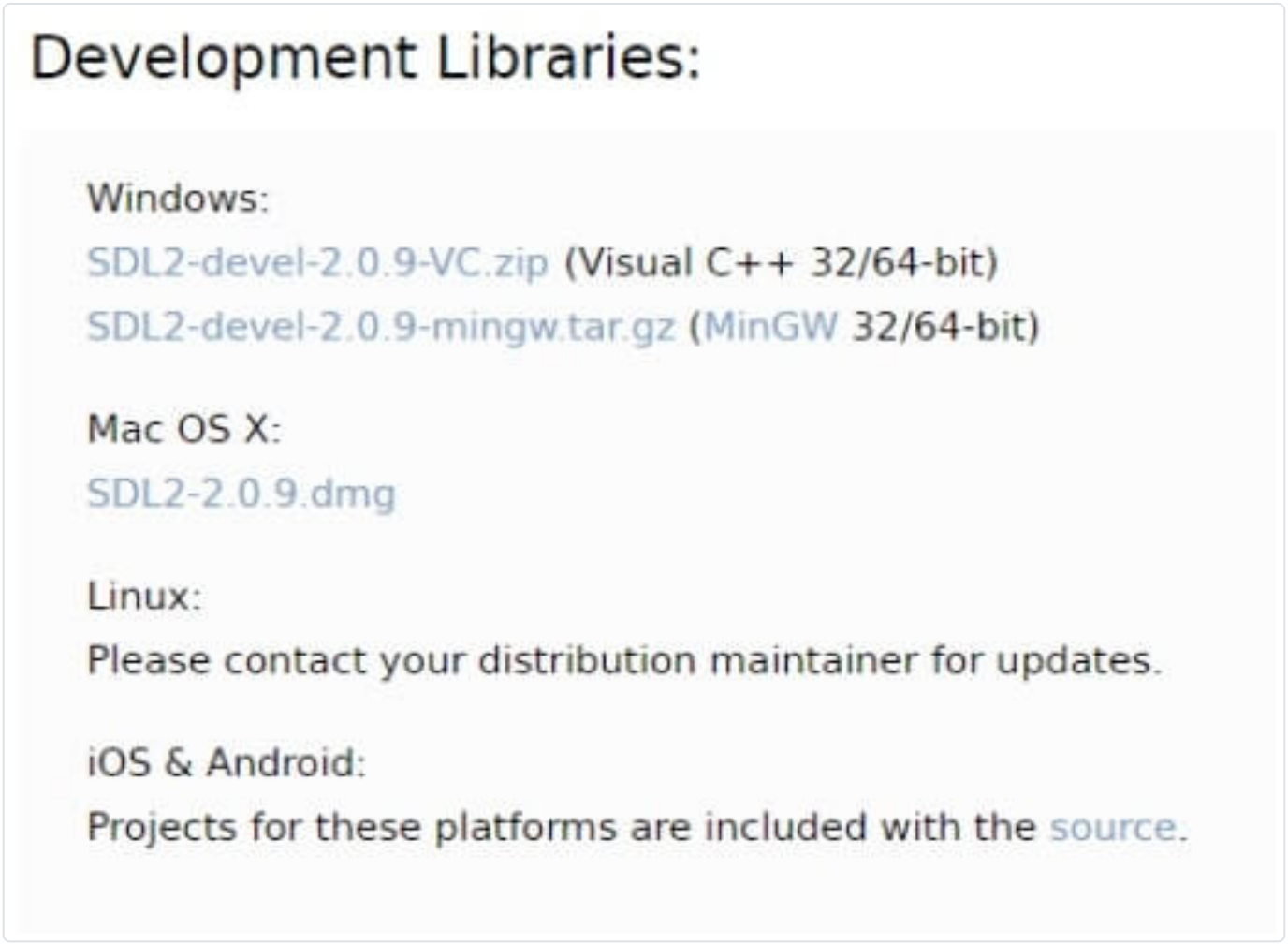
À gauche vous avez 3 menus de navigation (main, documentation, download).



Pour l'instant il n'y a que la partie download qui nous intéresse. Dans cette section vous avez :

- **SDL 2.0**: la dernière version SDL stable (c'est la version qu'on utilisera)
- **SDL 1.2**: l'ancienne version d'SDL (nous ne l'utiliserons pas)
- **SDL Mercurial**: la version en développement (nous ne l'utiliserons pas) c'est une version qui comporte les dernières fonctionnalités non stables de la bibliothèque. Elle permet donc de profiter des dernières mises à jour de la bibliothèque mais a contrario vous avez plus de risques de rencontrer des bugs.
- **SDL Bindings** : le portage de la SDL vers plusieurs langages de programmation comme le langage Go...

Il faut cliquer sur SDL 2.0 et vous tomberez sur une page de téléchargement, sur cette page vous retrouverez encore trois parties. Ici la partie qui nous intéresse est la partie **Development Libraries** :



Dans les prochains chapitres nous allons apprendre à installer la librairie et à compiler notre premier programme SDL 2.0 sous l'OS **Linux** et **Windows**.

Installation et compilation pour Linux [g++ et clang++] :

Installation

Pour Linux, la page de téléchargement nous informe que nous devons contacter notre distribution, pour installer la SDL 2.0. Nous utiliserons le gestionnaire de paquets qu'offre votre distribution. Généralement la SDL est disponible sur plusieurs distributions. Pour ce tutoriel on utilisera la distribution **Ubuntu** .

On va commencer d'abord par mettre à jour nos paquets :

```
sudo apt-get update && sudo apt-get upgrade
```

Par la suite, on va installer le paquet de développement SDL :

```
sudo apt-get install libsdl2-dev
```

Compilation

Voila vous avez installé la bibliothèque SDL sur votre OS Linux c'était **rapide et simple** , mais maintenant il va falloir créer un projet de test et par la suite le compiler pour vérifier ensuite si votre installation est bonne.

Commencez par créer un dossier `SDL` avec un dossier `bin` et `src` :

```
mkdir -p SDL/{src,bin}
```

- `bin` : on retrouvera l'exécutable au format ELF
- `src` : on retrouvera les fichiers source au format .cpp

Créer un fichier `main.cpp` qui contiendra le code source ci-dessous

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

int main(int argc, char* argv[])
{
```

```
    if(SDL_Init(SDL_INIT_VIDEO) < 0)
    {
        SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "[debug] %s", SDL_GetError());
        return -1;
    }
    SDL_Quit();
    return 0;
}
```

Il ne vous reste plus qu'à le compiler en partant du répertoire **SDL** en tapant la commande suivante :

```
g++ src/main.cpp -o bin/prog -lSDL2main -lSDL2
```

Si vous n'avez pas d'erreurs, c'est que votre installation c'est bien déroulée !

Installation et compilation pour Windows :

Installation

Pour Windows, la page officielle nous informe de deux possibilités :

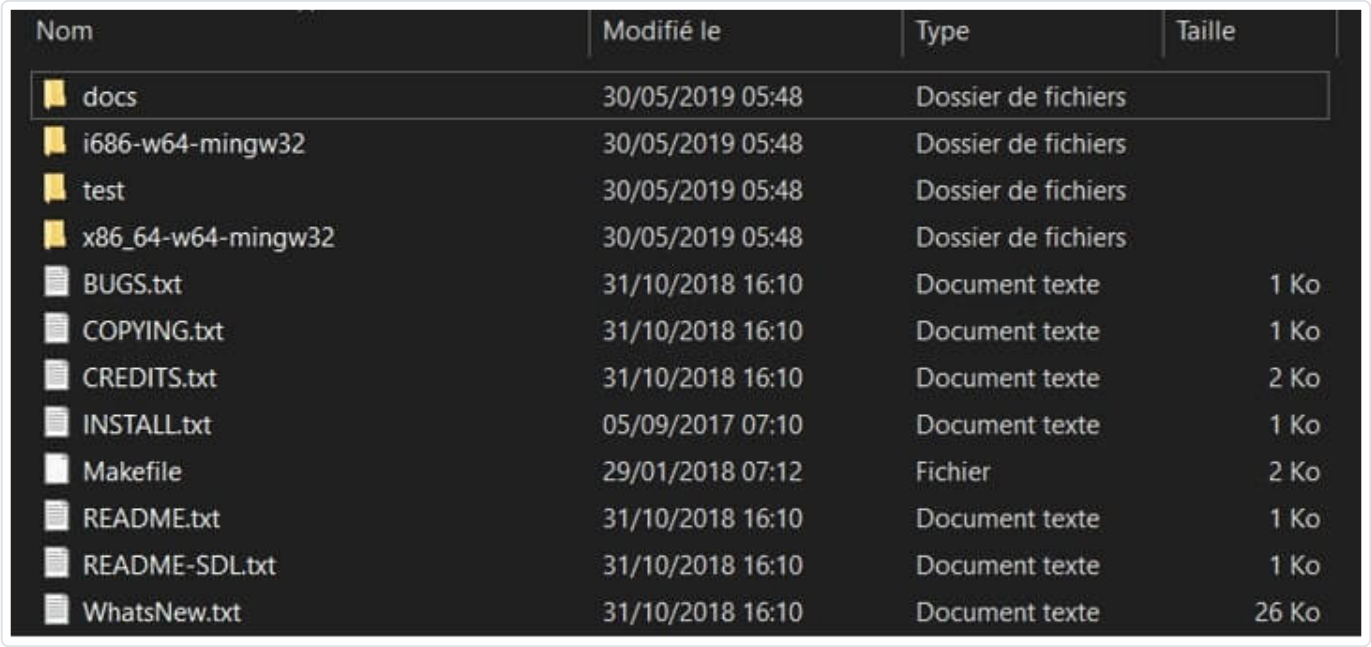
- SDL2-devel-2.0.9-VC.zip (Visual C++ 32/64-bit)
- SDL2-devel-2.0.9-mingw.tar.gz (MinGW

Nous utiliserons, le deuxième lien SDL2-devel-2.0.9-mingw.tar.gz (MinGW 32/64-bit), en cliquant dessus vous pouvez télécharger la SDL.

On va commencer par préparer, notre environnement de travail et par la suite on créera un répertoire qui contiendra SDL pour MinGW. Voici déjà à quoi ressemble mon répertoire de travail : **D:/SDL/MinGW** :

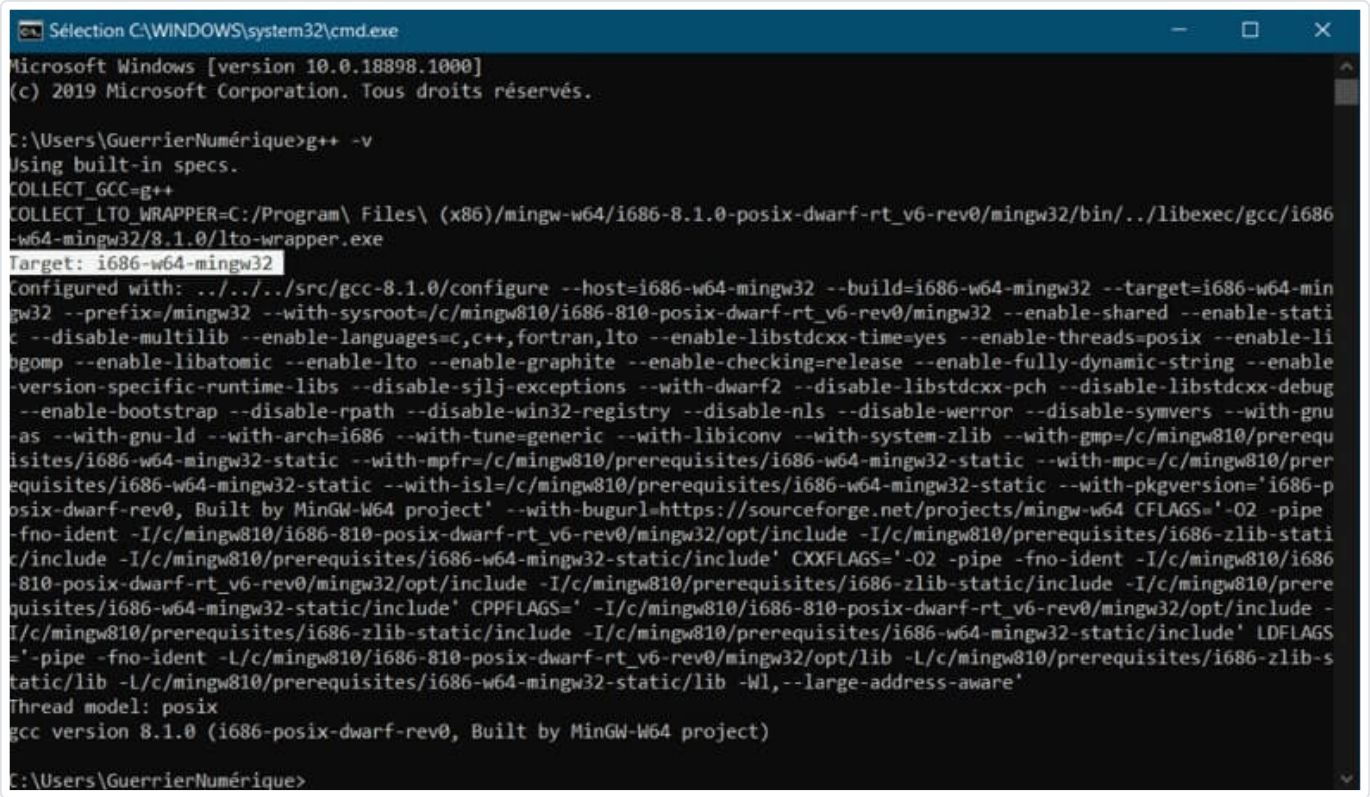
Dans ce répertoire je décompresse le tar.gz que j'ai téléchargé auparavant. Une fois cette étape finie, je me retrouve avec le répertoire **D:\SDL\MinGW\SDL2-2.0.9** et à

l'intérieur de ce répertoire j'obtiens l'arborescence suivante :



Nom	Modifié le	Type	Taille
docs	30/05/2019 05:48	Dossier de fichiers	
i686-w64-mingw32	30/05/2019 05:48	Dossier de fichiers	
test	30/05/2019 05:48	Dossier de fichiers	
x86_64-w64-mingw32	30/05/2019 05:48	Dossier de fichiers	
BUGS.txt	31/10/2018 16:10	Document texte	1 Ko
COPYING.txt	31/10/2018 16:10	Document texte	1 Ko
CREDITS.txt	31/10/2018 16:10	Document texte	2 Ko
INSTALL.txt	05/09/2017 07:10	Document texte	1 Ko
Makefile	29/01/2018 07:12	Fichier	2 Ko
README.txt	31/10/2018 16:10	Document texte	1 Ko
README-SDL.txt	31/10/2018 16:10	Document texte	1 Ko
WhatsNew.txt	31/10/2018 16:10	Document texte	26 Ko

Ici j'ai deux dossiers : `i686-w64-mingw32` et `x86_64-w64-mingw32`. Le premier dossier est pour un compilateur en **64 bits** et le second est pour un compilateur en **32 bits**. Pour savoir lequel choisir, il faut exécuter la commande `g++ -v` (ou `gcc -v` pour les développeurs C). Voici ce que vous devriez avoir :



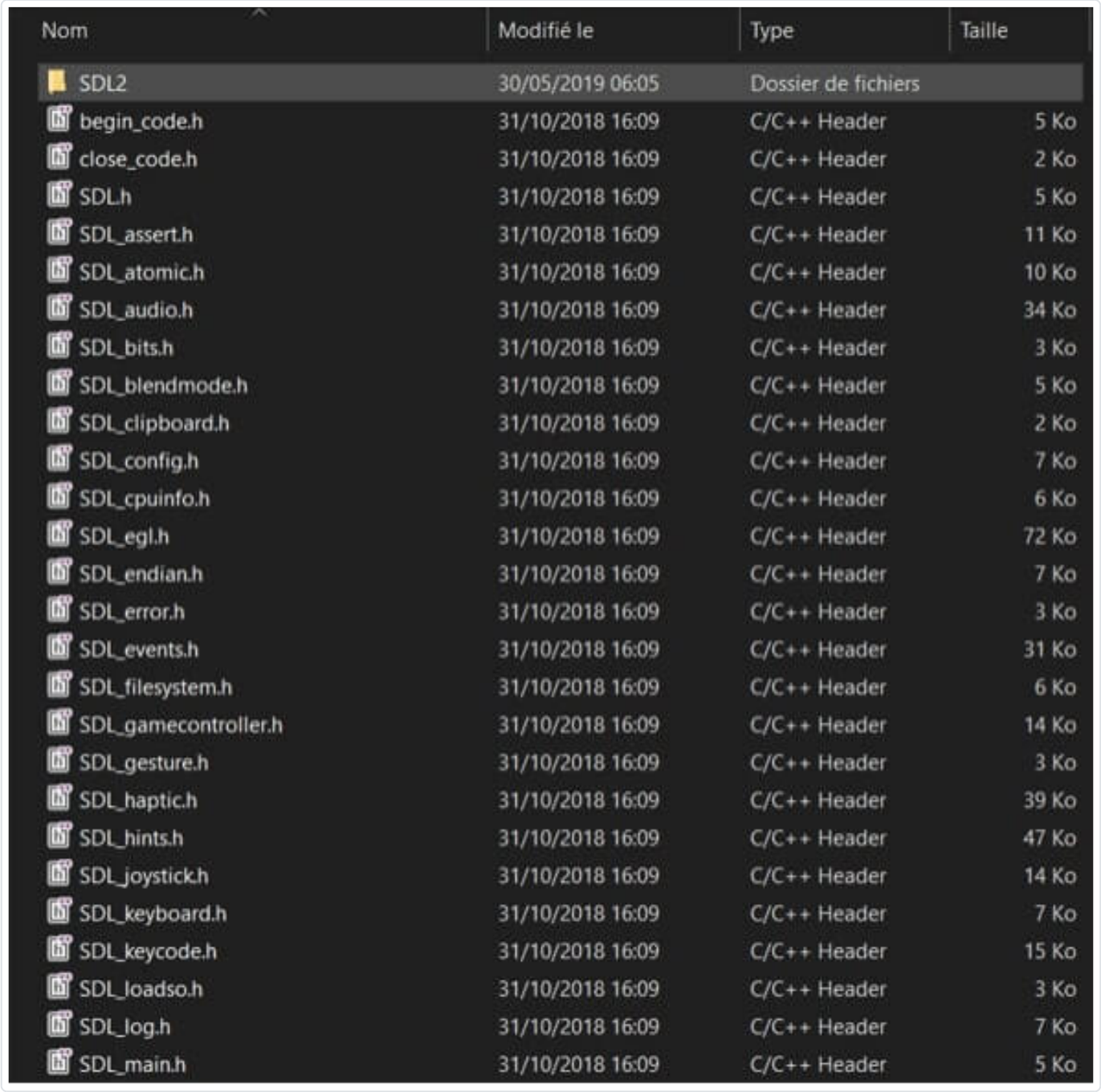
Ici je sais clairement que mon compilateur est pour le 64 bits, si vous avez autre chose, cela signifie que vous avez un compilateur 32 bits. Maintenant je sais que personnellement si je veux faire un projet SDL je vais utiliser celui-ci .

Ensuite, je vais créer un répertoire **SDL** et dans ce répertoire, je vais copier-coller, le contenu de mon dossier : **i686-w64-mingw32** (pour les compilateurs 32 bits ce sera **x86_64-w64-mingw32**

Je me retrouve ainsi avec les dossiers suivants dans le répertoire **SDL** :

Ce que je vous conseille de faire, c'est de modifier le dossier **include**, pour ne plus avoir de sous dossier **SDL2**. Nous aurons ainsi directement les headers dans le

dossier `include`. Si vous avez fait cette manipulation alors voici à quoi va ressembler ce dossier :



Nom	Modifié le	Type	Taille
SDL2	30/05/2019 06:05	Dossier de fichiers	
begin_code.h	31/10/2018 16:09	C/C++ Header	5 Ko
close_code.h	31/10/2018 16:09	C/C++ Header	2 Ko
SDL.h	31/10/2018 16:09	C/C++ Header	5 Ko
SDL_assert.h	31/10/2018 16:09	C/C++ Header	11 Ko
SDL_atomic.h	31/10/2018 16:09	C/C++ Header	10 Ko
SDL_audio.h	31/10/2018 16:09	C/C++ Header	34 Ko
SDL_bits.h	31/10/2018 16:09	C/C++ Header	3 Ko
SDL_blendmode.h	31/10/2018 16:09	C/C++ Header	5 Ko
SDL_clipboard.h	31/10/2018 16:09	C/C++ Header	2 Ko
SDL_config.h	31/10/2018 16:09	C/C++ Header	7 Ko
SDL_cpuinfo.h	31/10/2018 16:09	C/C++ Header	6 Ko
SDL_egl.h	31/10/2018 16:09	C/C++ Header	72 Ko
SDL_endian.h	31/10/2018 16:09	C/C++ Header	7 Ko
SDL_error.h	31/10/2018 16:09	C/C++ Header	3 Ko
SDL_events.h	31/10/2018 16:09	C/C++ Header	31 Ko
SDL_filesystem.h	31/10/2018 16:09	C/C++ Header	6 Ko
SDL_gamecontroller.h	31/10/2018 16:09	C/C++ Header	14 Ko
SDL_gesture.h	31/10/2018 16:09	C/C++ Header	3 Ko
SDL_haptic.h	31/10/2018 16:09	C/C++ Header	39 Ko
SDL_hints.h	31/10/2018 16:09	C/C++ Header	47 Ko
SDL_joystick.h	31/10/2018 16:09	C/C++ Header	14 Ko
SDL_keyboard.h	31/10/2018 16:09	C/C++ Header	7 Ko
SDL_keycode.h	31/10/2018 16:09	C/C++ Header	15 Ko
SDL_loadso.h	31/10/2018 16:09	C/C++ Header	3 Ko
SDL_log.h	31/10/2018 16:09	C/C++ Header	7 Ko
SDL_main.h	31/10/2018 16:09	C/C++ Header	5 Ko

Vous pouvez des à présent créer un dossier `src`, et par la suite créer un fichier `main.cpp` et de coller dessus le code source suivant :

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

```
int main(int argc, char* argv[])
{
    if(SDL_Init(SDL_INIT_VIDEO) < 0)
    {
        SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "[debug] %s", SDL_GetError());
        return -1;
    }
    SDL_Quit();
    return 0;
}
```

Comme pour Linux, il ne vous reste plus qu'à compiler votre code source en tapant la commande suivante :

```
gcc main.c -o prog -I include -L lib -lmingw32 -lSDL2main -lSDL2
```

Si vous n'avez aucune erreur, c'est que votre installation c'est bien passée !