

## Prise en charge des images en SDL ?

Nativement, la SDL ne permet la manipulation d'image qu'au format bmp. Il est donc nécessaire d'installer et de configurer l'API `SDL_image`.

Grâce à cette dernière, il sera possible de charger des images dans un projet en tant que surface et/ou texture.

## Installation et configuration de `SDL_image`

En supposant que Code :: Blocks et SDL2 soient installés correctement

- Installer

Ajoutez votre API en copiant les fichiers des dossiers `bin`, `lib`, `include` et `pkgconfig` de ce dernier dans les dossiers correspondants de `SDL2-[2.x.x]`.

Gardez votre explorateur de fichiers ouvert, on en aura besoin.

- Configurer dans Code :: Blocks

Ouvrez Code :: Blocks, allez dans `settings/compiler`. Une fenêtre apparaît.

Dans l'onglet *Linker settings*, ajoutez les chemins vers les fichiers lib dans l'encart de gauche `Link libraries`. L'ordre suivant est important !

```
\SDL2-2.0.3\i686-w64-mingw32\lib\libSDL2_image.a  
\SDL2-2.0.3\i686-w64-mingw32\lib\libSDL2_image.dll.a
```

Il s'agit des chemins que vous devriez avoir mais dans le cas contraire, reportez-vous à votre explorateur de fichiers pour les vérifier et les corriger.

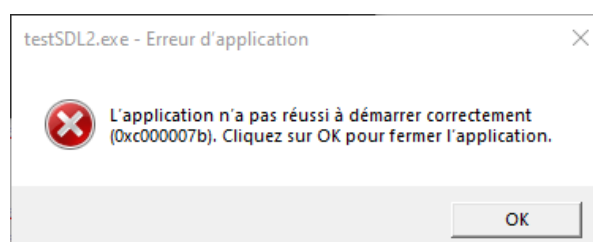
Ajoutez également dans l'encart de droite, *Other linker options*, les liens suivants.

- `-lSDL2_image`

- Installation terminée !

## Charger une image en SDL

Résultat systématique lors de mes tentatives :



## Le tiling et les tilesets

Un tileset est une grille d'images réunissant les assets d'un jeu, d'un niveau ou d'un événement.

Le tileset du tout premier Mario sur Nes par exemple (remixé par un certain Arrow)



L'utilisation des tilesets, c'est-à-dire le tiling, permet de simplifier certains codes, et également d'économiser les ressources du système puisqu'il n'aura pas à aller charger des images à plusieurs reprises.

A noter que dans le cadre des sprites de personnage et d'entités, on utilisera plutôt le terme de spritesheet.

## Les images animées

Résultat systématique lors de mes tentatives :

