Karnaj

Zeste de savoir

Utiliser la SDL sous Linux avec Code : :Blocks

20 janvier 2019

Table des matières

1. Introduction

J'ai déjà vu passer plusieurs messages de personnes se demandant comment utiliser la SDL 2 avec Code : :Blocks sous Linux. Bonne nouvelle, ce petit billet ~~vert~~ détaille rapidement comment le faire.

Pour commencer, il faut les fichiers nécessaires au bon fonctionnement de la SDL. Le mieux pour cela est d'utiliser son gestionnaire de paquets pour installer le paquet de développement de la SDL. Ils varient suivant la distribution, mais une petite recherche donne rapidement ce qu'il faut installer.

```
pacman -S sdl2
apt-get install libsdl2-dev
etc.
```

Le truc qui est bien avec ces paquets, c'est qu'ils placent directement les fichiers de la bibliothèque à un endroit approprié.

Ensuite, après avoir créer un projet console sous Code : :Blocks il ne reste qu'à lui indiquer qu'on veut qu'il lie la SDL au projet. Pour cela, dans les options du projet (« Project » → « Build options » → « Linker settings »), il faut donner à l'éditeur de lien (fenêtre « Other linker options ») les bonnes librairies à lier, c'est-à-dire ceci.

```
-lSDL2
```

Mais il y a mieux. En effet, `sdl2-config` permet d'obtenir ce qu'il faut pour la compilation. Ainsi, `sdl2-config --cflags` donne les options du compilateur (répertoire des fichiers d'en-tête) et `sdl2-config --libs` donne les bibliothèques à lier. Nous pouvons par exemple demander à les afficher dans un terminal avec `echo`.

```
$ echo $(sdl2-config --libs)
$ echo $(sdl2-config --cflags)
```

Les `$()` sont là pour indiquer au terminal qu'il ne faut pas considérer ce qui est écrit comme du simple texte, mais qu'il faut l'interpréter (par exemple avec `echo $(echo 25)`, on aura `25` alors qu'avec `echo echo 25`, on obtient `echo 25`.

Ainsi, dans la fenêtre « Project » → « Build options » → « Linker settings » → « Other linker options », nous mettrons `$(sdl2-config --libs)` et dans la fenêtre « Project » → « Build options » → « Compiler settings » → « Other compiler options », nous mettrons `$(sdl2-config --cflgas)`.

Compiler à la main

Bien sûr, `sdl2-config` nous permet également de compiler à la main. Par exemple, si nous voulons compiler un fichier `main.c`.

```
1 gcc main.c $(sdl2-config --cflags --libs)
```