SDL2 - Questionnaire

1/ La SDL2 est un ensemble de dossiers regroupent des fichier de codes

(en ce cas du C) qui facilite la programmation d'éléments visuel et

de calcules pour des programmes plus compliquées (Comme des jeux vidéos) en 2D

2/ La SDL2 peux permettre de faire :

- Donne une douzaines de fichiers.h avec des éléments de programmations

(Calcules, Fonctions...)

- Aide à la réalisation de jeux en 2D (Platforme, Tir, Aventure, Text...)

- Des dessins sur l'écran, capable de les affichers

- Conteneur vidéos 2D

3/ SDL2 Fonctionne sur :

- Windows CE

- BeOS (SDL 1.2)

- Mac OS (SDL 1.2)

- OS/2 (SDL 1.2)

(UNIX)

- Mac OS X

- Linux

- Android

- BSD (FreeBSD, NetBSD, OpenBSD, BSD/OS)

- Solaris

- IRIX

- QNX (SDL 1.2)

4/ Installation :

- Il faut se rendre sur le site de SDL (http://www.libsdl.org/index.php)

- Se rendre sur la page des sorties des versions SDL (Download - SDL Releases)

- Après il faut téléchanger la dernière version de la bibliothèque disponible

(Il y as Source code pour la manipulation du code lui-même, Developement Libraries ce qui donne les versions pour développeurs (et nous aussi) et Runtime Binaries)

(Le fichier "SDL2-2.0.1-win32-x86.zip" correspond à une version 32 bits contrairement à l'archive "SDL2-2.0.1-win32-x64.zip" qui correspond à une version 64 bits)

(Pour des raisons de comptabilité, on télécharger la version 32 bits)

- Une fois le téléchargement fini, extraire les fichiers (au préalable dans un dossier) dans un dossier proche de ceux avec lesquelles ont veut utiliser pour travailler sur notre code

Configuration :

- Lors de la création d'un projet (Codeblock par exemple), il faut créer un "Empty Project" pour y associer les dossiers SDL2 (Ne pas faire un projet "SDL" ce dernier utilisera une version 1.2)

- Cliquer sur "projet" puis "Build option" , onglet "Linker settings" pour pouvoir ajouter (ou lier) les fichiers "libSDL2main.a" et "libSDL2.dll.a" sur Link libraries

(Vous pouvez choisir de "Keep the path relative" si vous le désirez)

- Puis sur "Other link option", veuillez ajouter "-lmingw32"

- Allez sur "Search directories" et ajoutez le dossier "include" et "lib"

- Pour finir, déplacer le fichier SDL2.dll dans le dossier avec le reste de votre projet

(Vous pouvez également faire une arboréscence du projet si vous souhaitez le partager en ligne)

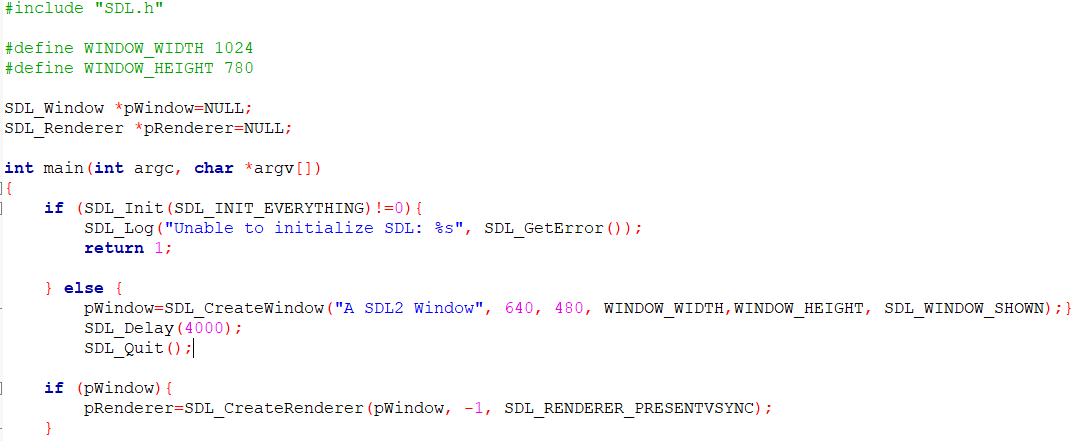
5/ La SDL2 apporte de nombreuses nouveautés et est bien plus performante que la SDL1(1.2)

* La SDL2 utilise le hardware, il pioche les ressources directement
* La SDL1.2 utilise des conteneurs, simplement

6/ Pour utiliser la SDL2 dans le projet, il faut l'initialiser;

- Ajouter <SDL2/SDL.h> pour initialiser un chemain pour utiliser la librarie de fonctions SDL2

- Remplacer int main par int main(int argc, char \*argv[]) pour utiliser des éléments de SDL.h pour la programmation

7/ Je vais expliquer comment créer une fenêtre en SDL2 avec le code si dessous

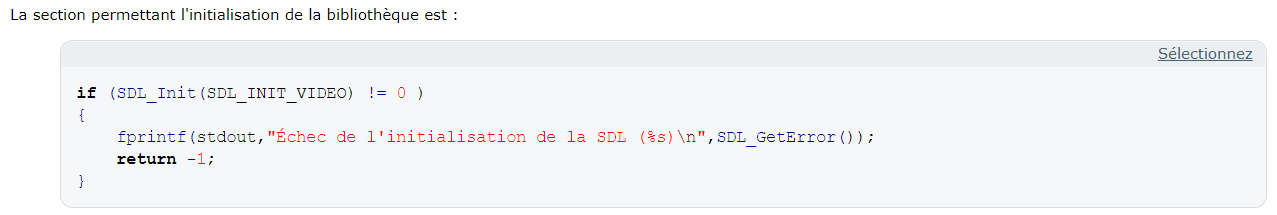
Pré-réquis :

* Inclure le headder SDL.h qui nous donne accès à des codes qui vont aider la réalisation de la fenêtre
* Définir la taille de la fenêtre avec une constante WINDOW\_WIDTH et WINDOW\_HEIGHT

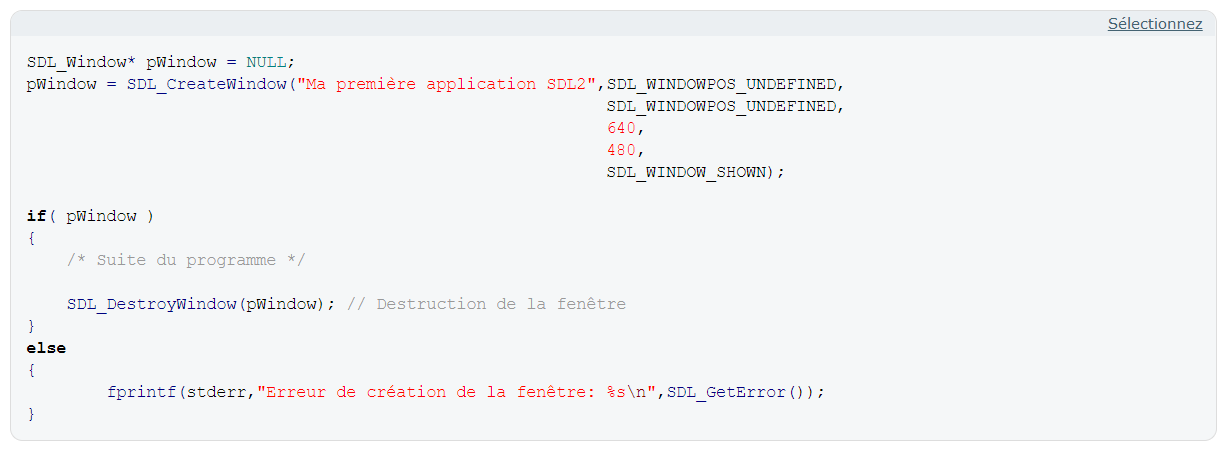
Procédure :

* On initialise des pointeurs (p) de variables (SDL\_Window, SDL\_Renderer) avec une valeur NULL (zéro, néant) pour qu’ils puissent être utilisable
* J’ai ajouté une condition if au cas si il y as un soucie de paramétrage, le cas échéant ELSE, il va créer la fenêtre avec le pointeur pWindow et la suite d’instruction SDL\_CreateWindow qui contient les informations nécessaires à la réalisation de la fenêtre

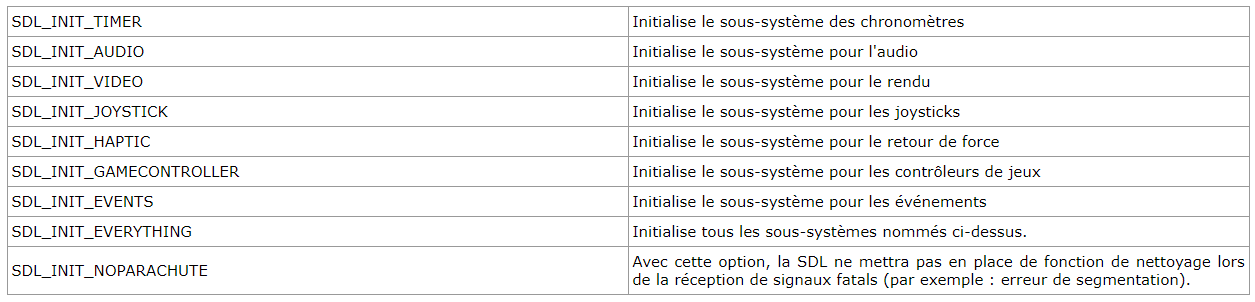
8/ Les primitives ;



Le code pour ouvrir une fenêtre :



La fonction avec primitive qui ouvre une fenêtre est ‘’SDL\_CreateWindow()’’ , ont peut ensuite utiliser de nombreuses primitives pour manipuler cette fenêtre avec le code :



Mais pour créer cette fenêtre, SDL attend des paramètres importants (indispensable) pour la créer, telle que : Hauteur, Longueur, Coordonnées X et Y, Nom et Options…

9/ Avec la SDL, le renderer est contenu dans une structure appelée **SDL\_Renderer**. Elle contient les informations du contexte d'affichage de la fenêtre.

Ainsi, pour créer un renderer, nous devons en tout premier lieu déclarer un pointeur de SDL\_Renderer. La raison de la déclaration d'un pointeur plutôt que la structure directement est purement pratique : toutes les fonctions de la SDL utilisent des pointeurs de renderer et non la structure en elle-même pour des raisons d'optimisation.

En effet, il est moins long de passer en argument une adresse plutôt qu'une structure SDL\_Renderer assez lourde.

SDL\_Renderer\* SDL\_CreateRenderer(SDL\_Window\* window, int index , Uint32 flags);

Tout d'abord, on remarque que cette fonction utilise un pointeur ‘’SDL\_Renderer’’, ou NULL en cas d'erreur.