# SDL2 Partie 1

## Qu’est-ce que la SDL2 ?

La sdl2 est la version 2 de Simple DirectMedia Layer qui est une bibliothèque logicielle libre graphique. Elle permet de gérer l’audio, les évènements, les affichages vidéo.

## Que peut faire la SDL2 ?

La sdl2 peut en autre être utilisée pour réaliser des applications multimédias en deux dimensions comme des jeux vidéo par exemple.

## Sur quelles plateformes fonctionnent la SDL2 ?

La sdl2 est multi-plateforme : Windows, Linux, Mac os, Android…

## Comment obtenir et installer la sdl2 sur un EDI ? Donnez les étapes de la configuration d’un projet sdl2 avec codeblocks (gcc).

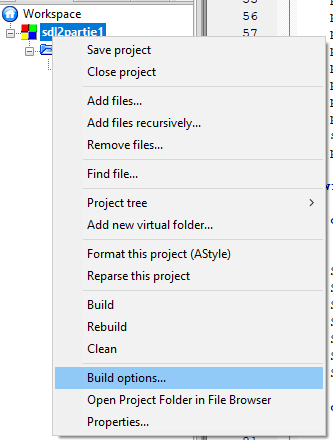
Vous devez vous rendre sur le site : <https://www.libsdl.org/download-2.0.php>

Puis télécharger le fichier : SDL2-devel-2.0.12-mingw.tar.gz

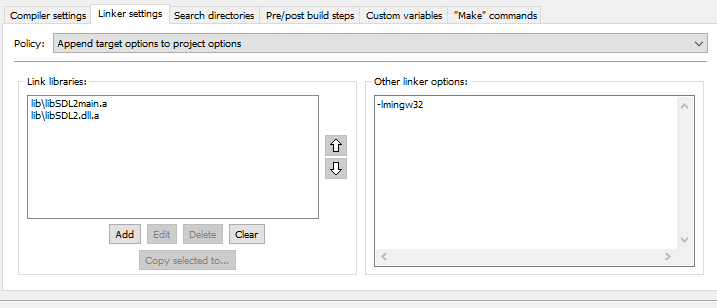
Ensuite il vous suffit d’extraire les sous-dossiers du fichier ZIP dans votre projet. C’est-à-dire : il faut extraire les deux sous-dossiers « lib » et « include » dans la racine de votre projet SDL2.

Ensuite il vous suffit d’extraire le fichier « SDL2.dll » présent dans le sous-dossier « bin » du fichier zip dans la racine de votre projet.

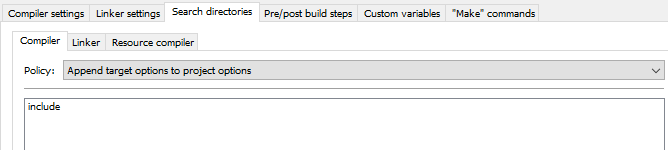
Ensuite dans codeblock il vous suffit de vous rendre dans le « build option » de votre projet :



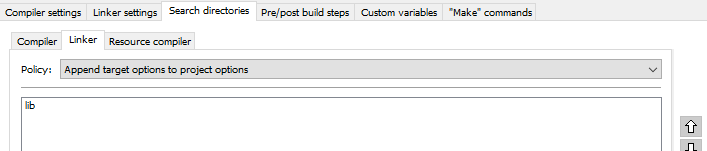
Puis dans « linker setting » il vous suffit d’indiquer les différentes lignes présentes sur le screen suivant :



Ensuite dans « search directories » dans l’onglet « compiler » vous devez ajouter le dossier « include » précédemment extrait comme le screen ci-dessous :



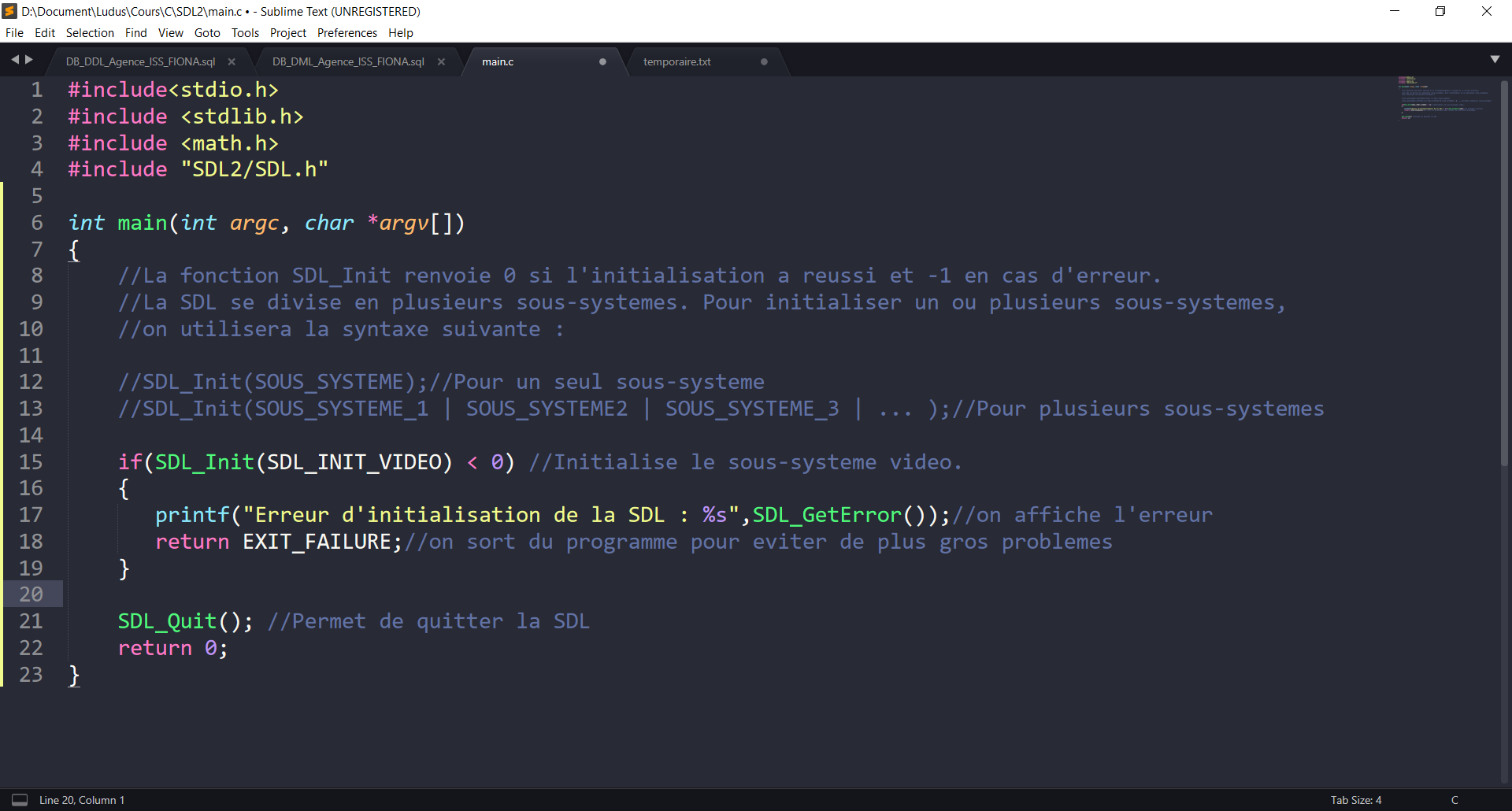
Et enfin d’ans l’onglet « Linker » toujours dans « search directories » vous devez ajouter le dossier « lib » précédemment extrait comme le screen suivant.



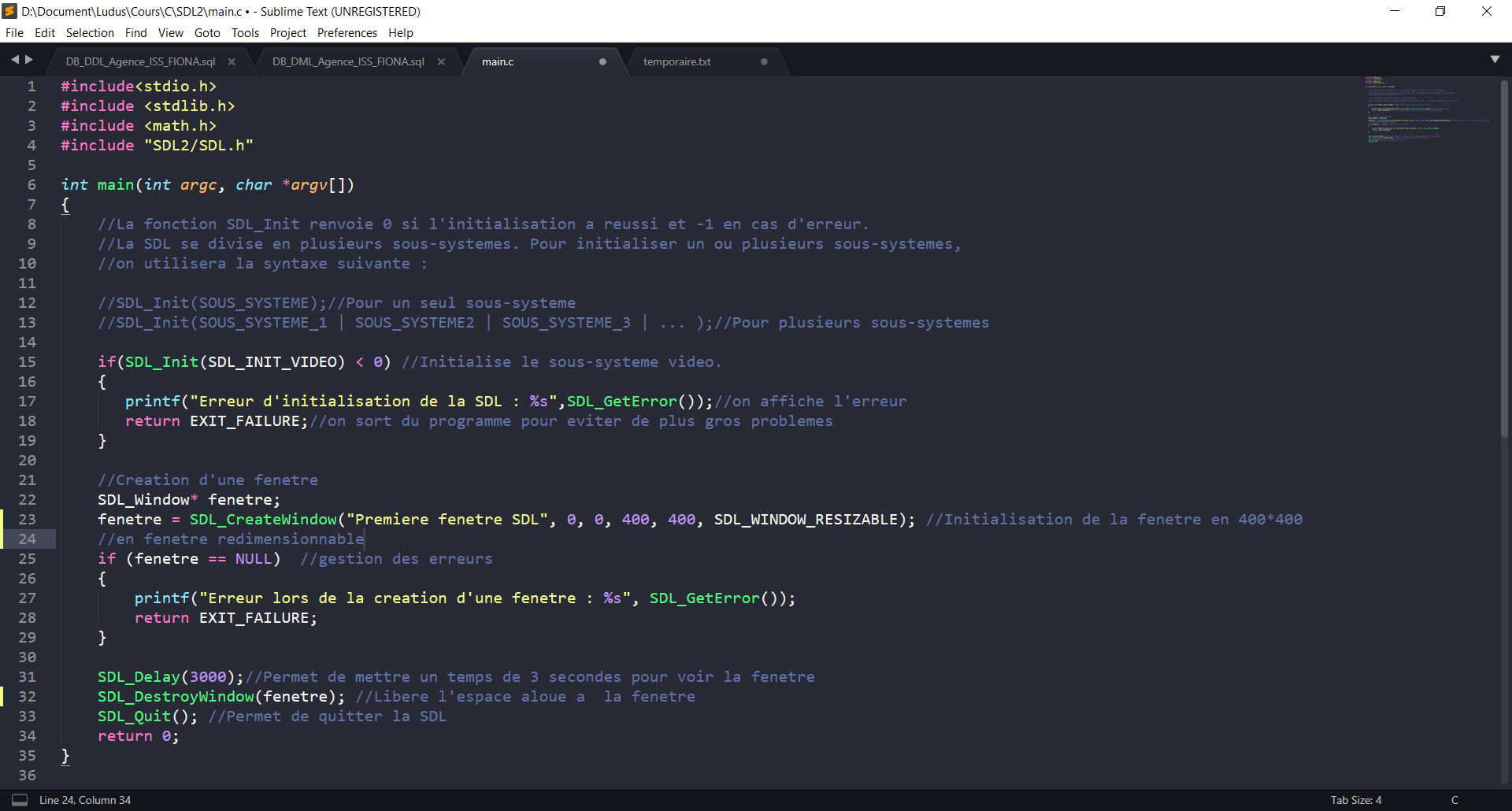
## Quelles sont les différences entre la sdl1 et sdl2 ?

La sdl2 est la version mise à jour de la sdl1 donc c’est une version améliorée. Elle accueille aussi de nouvelles fonctionnalités telles que : La prise en charge de plusieurs fenêtres, plusieurs écrans, plusieurs périphériques audios etc.

## Pour utiliser la SDL2 dans votre projet, vous devez initialiser l’instance sdl2, quelles sont les étapes d’initialisation à réaliser pour utiliser la lib sdl2, donnez le code correspondant et expliquer précisément chaque instruction.



## Comment créer une fenêtre avec sdl2, donnez et expliquez le code correspondant à la création d’une fenêtre avec sdl2.

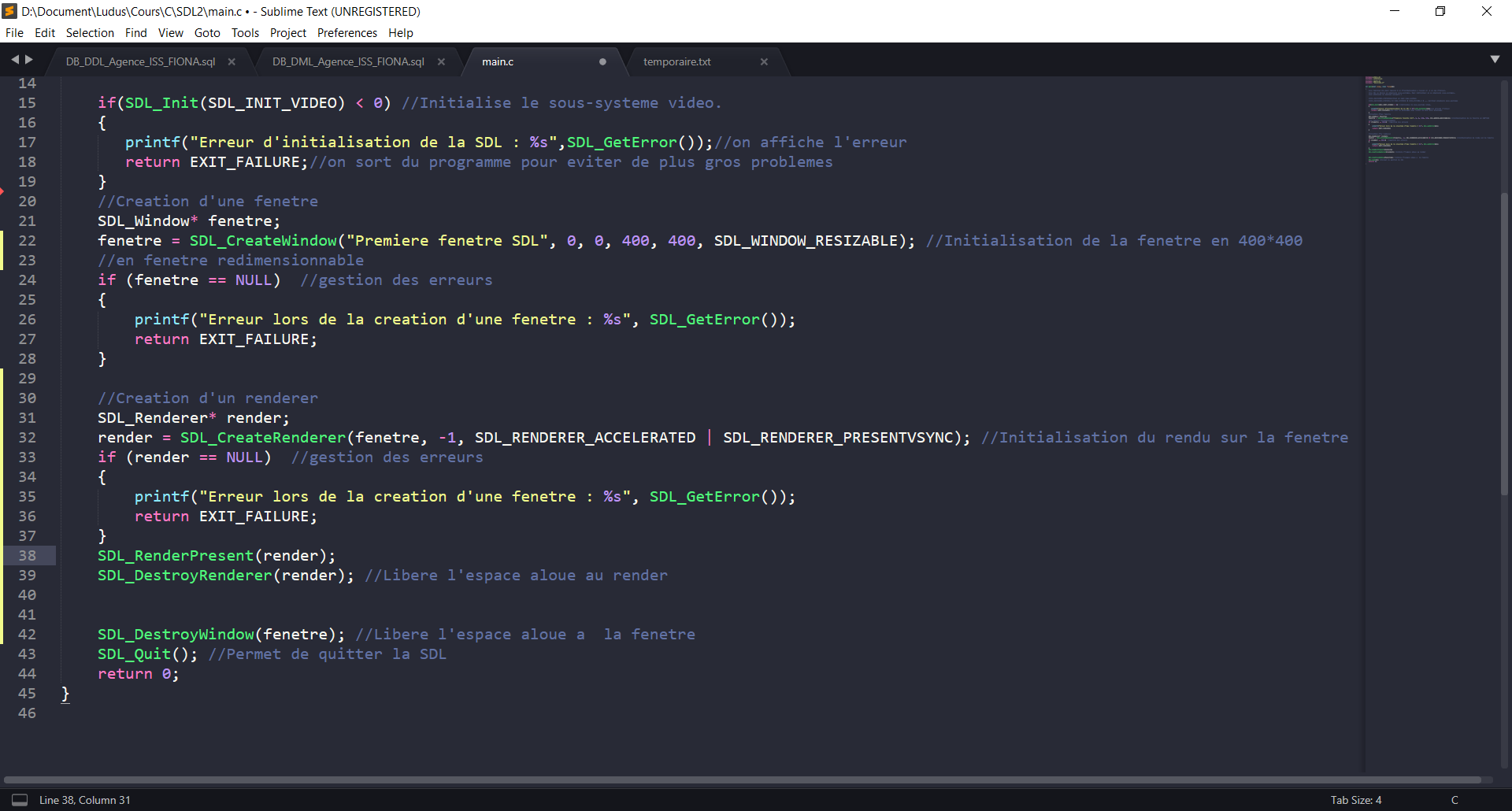


## Gestion d’une fenêtre sdl2, donnez les primitives associées à la fenêtre sdl2.

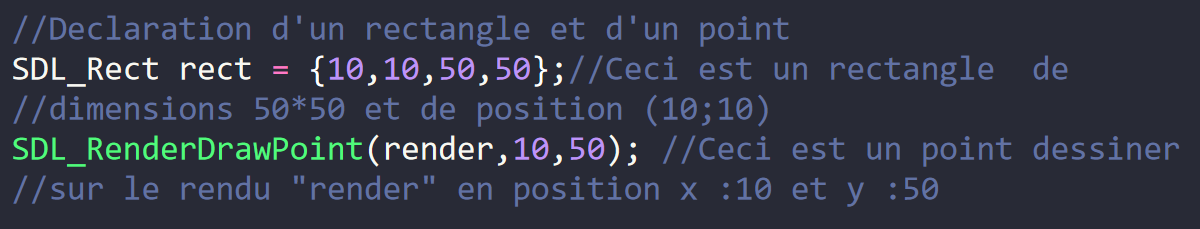
|  |
| --- |
| SDL\_SetWindowTitle(SDL\_Window\* window, const char\* title); |
| SDL\_SetWindowIcon(SDL\_Window\* window, SDL\_Surface\* icon); |
| SDL\_SetWindowPosition(SDL\_Window\* Window, int x, int y); |
| SDL\_GetWindowPosition(SDL\_Window\* Window, int x, int y); |
| SDL\_DestroyWindow(SDL\_Window\* Window); |
| … |

## Gérer un rendu : qu’est-ce qu’un « renderer », comment est-il associé à la fenêtre sdl2 ? Donnez et expliquez le code permettant de gérer un rendu, création, affichage, effacement…

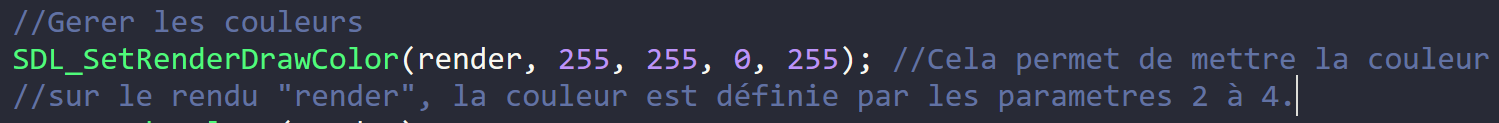
Le renderer est en fait l'espace où nous allons pouvoir dessiner, écrire... Or, pour qu'un dessin ou un mot s'affiche à l'écran, il faut faire un rendu, c'est-à-dire créer l'image finale que l'on veut afficher, puis l'afficher.



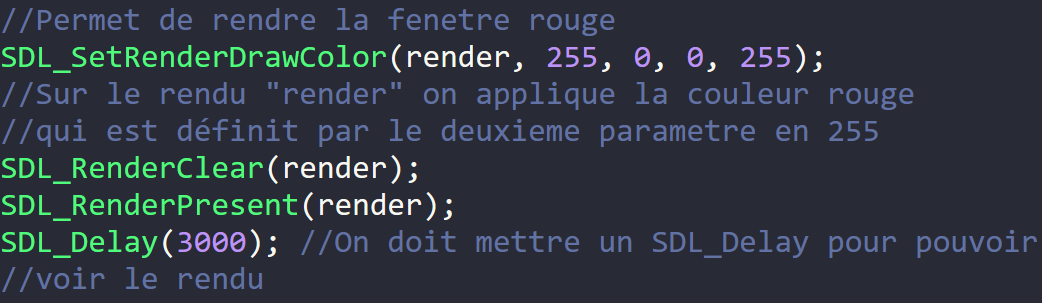
## SDL\_Point et SDL\_Rect : Donnez et expliquez le code de la déclaration d’un SDL\_Point et SDL\_Rect



## Les couleurs avec la sdl2 : donnez les instructions permettant de gérer la couleur avec la sdl2.



## Donnez le code permettant d’afficher un fond rouge dans le rendu.



## Dessiner dans le rendu : donnez le code des fonctions permettant de dessiner dans le rendu les formes suivantes : - carré vide

## - Carré plein

## - Cercle vide

## - Cercle plein

## Quelles sont les fonctions permettant de dessiner des points et des lignes ?

Pour dessiner des points, la fonction est : SDL\_RenderDrawPoint(SDL\_renderer\* renderer, int x, inty) ;  
Pour dessiner des lignes, la fonction est : SDL\_RenderDrawLine(SDL\_Renderer\* renderer, int x1, int y1, int x2, int y2) ;

## A quoi servent les fonctions SDL\_RenderClear et SDL\_RenderPresent ?

SDL\_RenderClear : Cette fonction permet d’effacer la ciblede rendu actuelle avec la couleur du dessin.

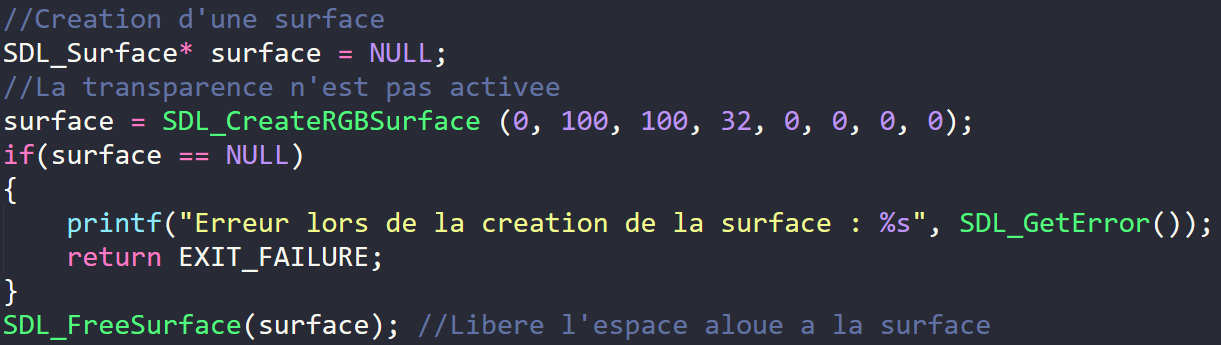
SDL\_RenderPresent : Cette fonction permet de mettre à jour l’écran avec tout rendu effectué depuis l’appel précédent.

## Expliquez SDL\_Delay.

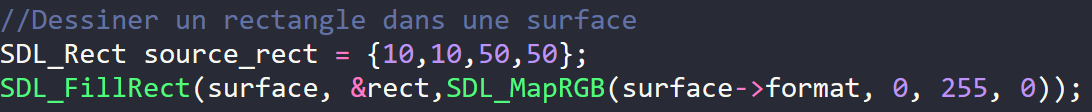
La fonction SDL\_Delay permet de réaliser une attente d’un nombre spécifié de millisecondes avec de revenir.

1. Qu’est-ce qu’une surface en sdl2.  
   La surface est l’équivalent des textures sauf qu’elle ne dispose pas de l’accélération matérielle et qu’elle n’a pas autant de possibilité de dessin que les textures.

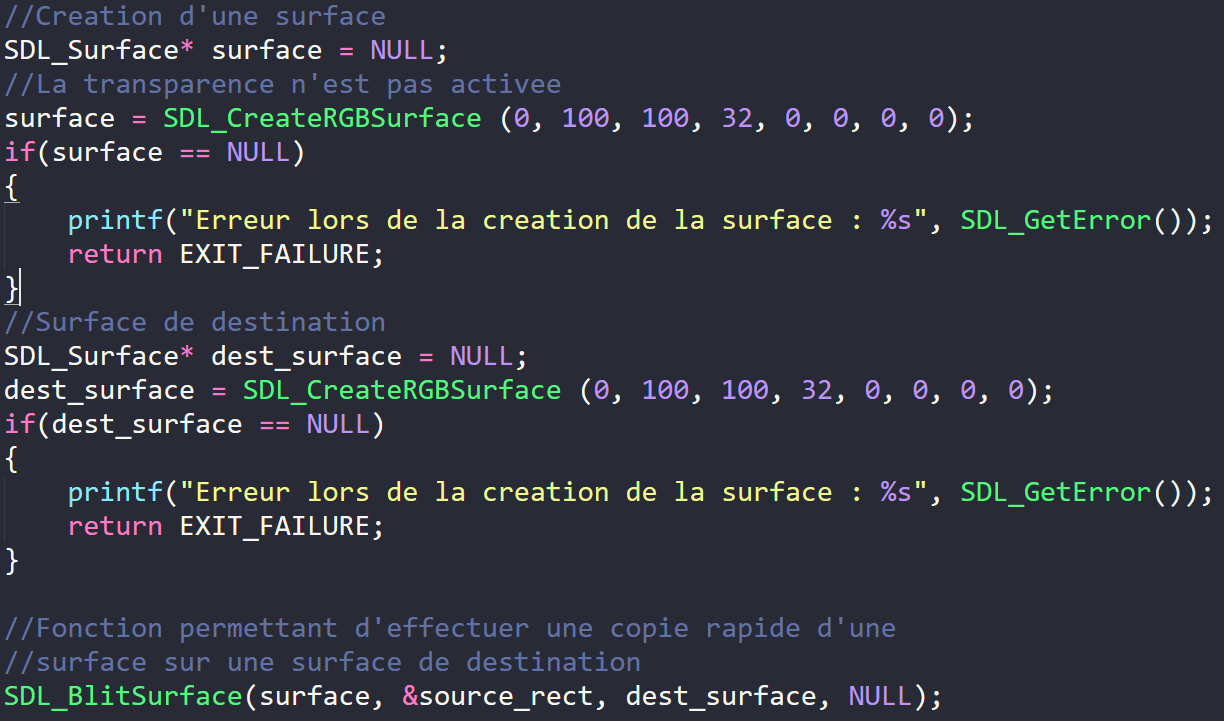
## Donnez le code permettant de créer une surface.



## Dessiner dans une surface : donnez le code de la fonction SDL\_FillRect.

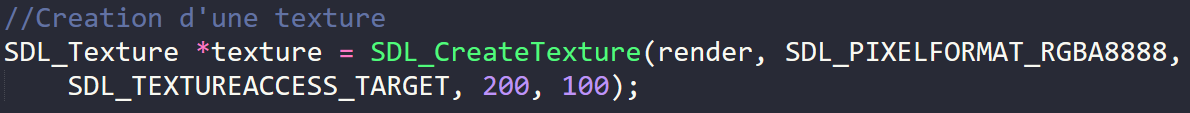


## SDL\_BlitSurface(..) : Donnez le code pour tester cette méthode.

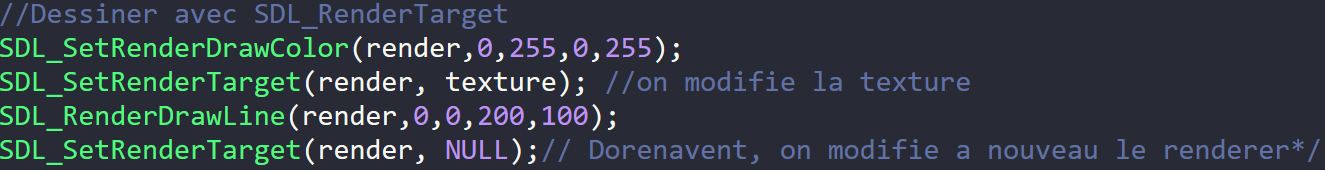


## Les textures : Qu’est-ce qu’une texture ? Donnez le code permettant de créer une texture.

Une texture, à l’aide des évenements, pourra déplacer des éléments sur l’écran, en faire disparaître ou apparaître et tout ça sans avoir à tout redessiner.

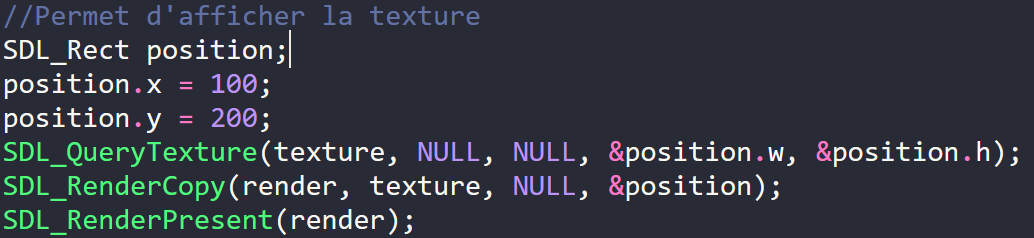


## Dessinez dans une texture : SDL\_RenderTarget, donnez le code pour tester cette méthode.



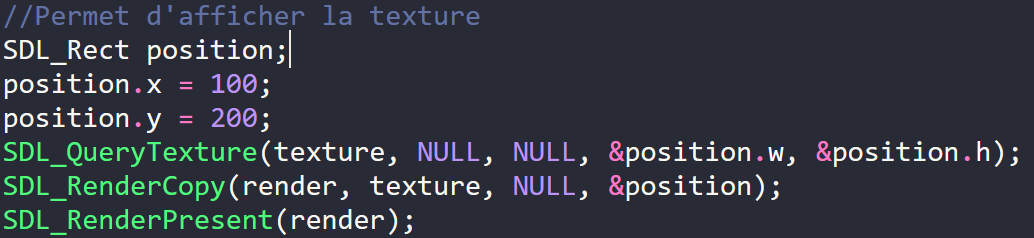
## 23)SDL\_RenderCopy : A quoi sert cette méthode, donnez le code pour tester cette méthode.

Cette fonction sert à copier une partie de la texture vers la cible de rendue actuelle.

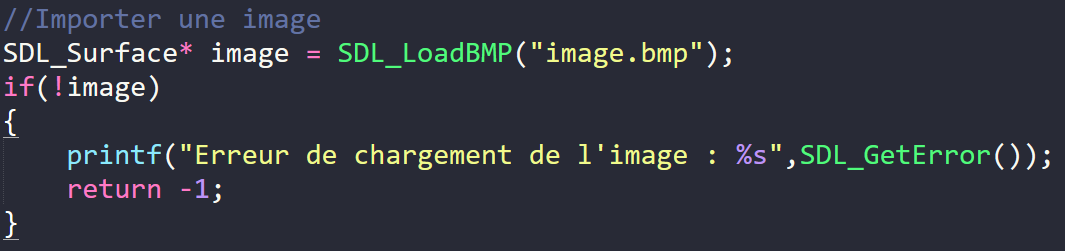


## 24)SDL\_QueryTexture : A quoi sert cette méthode, donnez le code pour tester cette méthode.

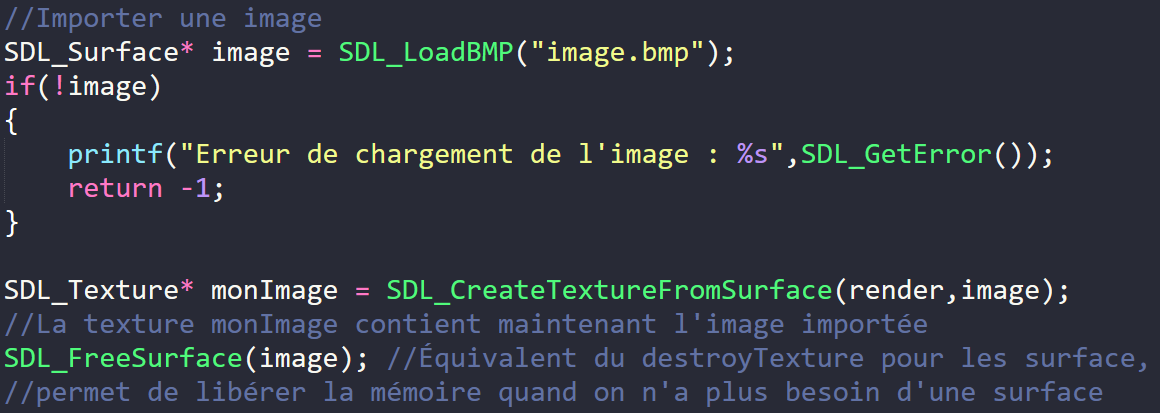
Cette fonction permet d’interroger et de récupérer les attributs d’une texture.



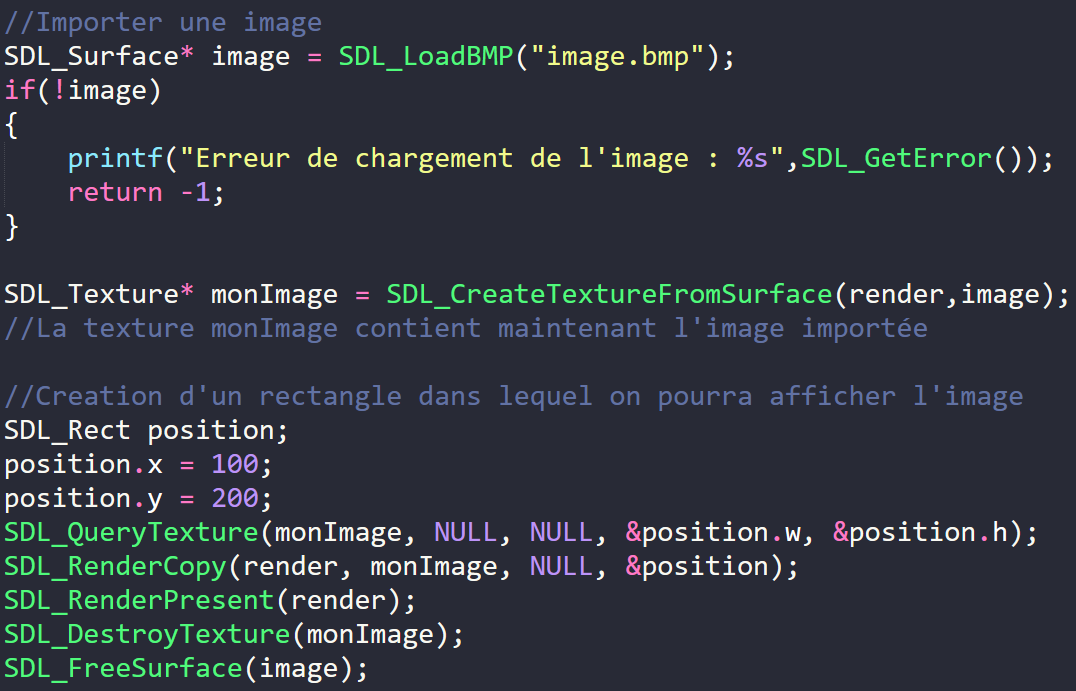
## 25)Les images : La sdl2 ne prend en charge nativement que les formats bmp, si vous souhaitez utiliser d’autres formats vous devrez associer la lib sdl2\_image à votre projet. SDL\_LoadBMP, est la méthode permettant d’utiliser des bmp, donnez le code permettant de tester cette fonction.



## 26)Donnez le code permettant de créer une texture à partir d’une image (bmp)



## 27)Prenez une image de votre choix au format bmp et testez les fonctions précédentes en ajoutant le code permettant d’afficher dans le rendu.



## 28)Depuis le début nous créons des instances permettant de manipuler la sdl2, cependant nous oublions une étape essentielle dans son utilisation, la destruction des instances, faites un relevé des instances crées et identifier les méthodes que vous devez utiliser pour libérer celles-ci.

|  |  |
| --- | --- |
| Instances : | Destruction |
| Fenêtre | SDL\_DestroyWindows(fenetre) |
| Rendu | SDL\_DestroyRenderer(rendu) |
| Surface | SDL\_FreeSurface(surface) |
| Texture | SDL\_DestroyTexture(texture) |

## 29) typedef struct sdl\_manager { SDL\_Window \*pWindow SDL\_Renderer \*pRenderer SDL\_Texture \*ptexture SDL\_Surface \*psurface } Vous avez codé votre programme, foncions et procédures à la volée afin de tester plus précisément les méthodes. A l’aide du type structuré ci-dessus, je vous demande de reprendre votre programme afin de l’améliorer et de proposer suivant les règles de la programmation modulaire et des notions de patterns (MVC) une nouvelle version plus structurée et plus performante que la version précédemment proposée.