**SDL 2 What is that ?**

**Partie 2**

# Quel Api faut-il ajouter à votre projet de développement afin de pouvoir prendre en charge les images au format (JPG, PNG…), que fait cet API ? (10 points)

Il faut ajouter la SDL\_Image à notre projet de développement pour pouvoir prendre en charge les images aux formats JPG, PNG et autre, cet API permet de prendre en charge ces formats d’images sans avoir à coder tous les algorithmes de chargement et conversion de ceux-ci.

# Donnez les étapes de configuration pour l’utilisation de ce nouvel API. (10 points)

Pour utiliser ce nouvel API il faut d’abord le téléchargé à l’adresse suivante : <https://www.libsdl.org/projects/SDL_image/release/SDL2_image-devel-2.0.5-VC.zip>

Ensuite, fusionner ces documents avec ceux de la SDL actuelle.  
Veillez à vérifier la cohérence des versions.

Après il faut récupérer les six dll suivants :

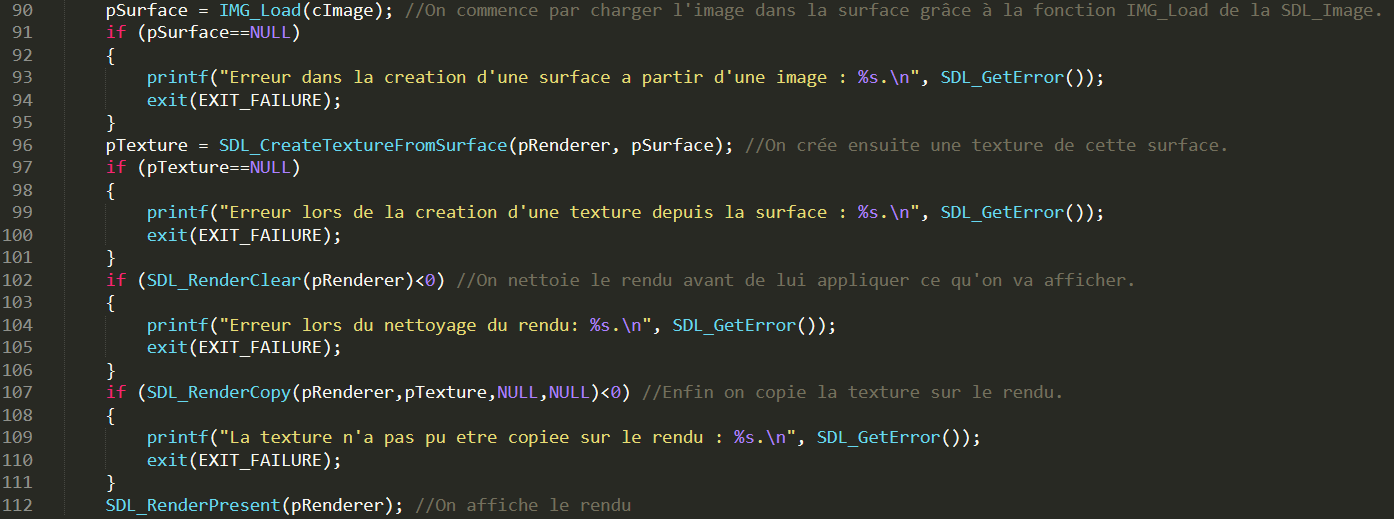


Et les placer à la racine du projet.  
Après il faut rajouter dans les options de compilation, les paramètre du linker les fichiers suivants : libSDL2\_image.a et libSDL2\_image.dll.a

Enfin il suffit de rajouter cette ligne dans le fichier d’entête : #include "SDL2/SDL\_image.h"

Et à partir de maintenant vous pouvez utiliser la SDL\_Image !

# Donnez un exemple de code en c chargeant une image au format png dans la fenêtre. Commentez et expliquez ce code. (10 points)



Pour l’exemple, voir le code du programme ci-joint.

# Les images animées : Gestion des Tileset. Expliquez ce qu’est un tileset, à quoi cela peut servir ? (10 points)

Un tileset est une image comportant divers « tile » : images de mêmes tailles qui vont composer le jeu, cela permet en jeu d’afficher un décor d’un niveau entier avec juste une petite image qui contient les éléments du niveau, ou bien d’avoir un personnage et toutes ses animations en une seule image.

# A partir de ces sites : <https://itch.io/game-assets/free/tag-tileset> , <https://opengameart.org/content/orthographic-outdoot-tiles> ou du site de votre choix, proposant des tilesets d’images libres de droit, créez une image animée dans votre fenêtre. (Vous simulez les changements d’animation à chaque frame). (15 points)

Pour l’exemple, voir le code du programme ci-joint.

Note : Ajout des évènements et du taux de rafraichissement à 60 secondes :

Cliquez pour fermer la fenêtre, déplacez votre souris pour déplacer l’image animée.

# Le Tile Mapping

# Expliquez ce qu’est le tile mapping, technique de l’image fixe ou tile mapping ? Comment contrôler les ressources lorsque l’on manipule des images dans le cadre de la construction des maps de jeux. (10 points)

Le Tile Mapping est la création d’un niveau de jeu avec une image tileset et un tableau d’entiers désignant quel morceau de l’image on veut à tel endroit du niveau.

Au contraire, la technique de l’image figée revient à créer une image complète formant tout le fond du niveau et l’appliquer sur l’écran. Cependant cette technique nécessite de stocker et charger une image assez lourde et empêche de différencier aisément dans notre programme les viers éléments du décor.

Il est donc préférable d’avoir une petite image tileset comportant des blocs de même taille à placer en fonction d’un tableau d’entier, que d’avoir une grosse image qui prend déjà de la place et qui est difficile à manipuler dans le programme.

# Existe-t-il des logiciels dédiés à la gestion des tiles, si oui nommez-en et expliquez leur fonctionnement. (10 points)

Il existe des logiciels à la gestion des tiles comme par exemple : TIled, MapWin, Ogmo Editor…

Ils demandent tout d’abord un tileset, puis à partir de là proposent de créer une map.  
Il faut renseigner la taille d’une case dans le tilemap, ainsi que les marges et espace si il y en a.  
Après cela, pour le logiciel Tiled par exemple, deux fichiers sont créés : un fichier .tsx comprenant les informations du tileset et un fichier .tmx qui va enregister la map créée.

Lien pour télécharger Tiled : <https://www.mapeditor.org/>

Pour l’exemple regarder les fichiers ci-joint : nature-paltformer-tileset-16x16.tsx et Map.tmx