

1/ Installer la bibliothèque SDL sous Code:Blocks

Tout d'abord, installer Code:Blocks. Puis allez sur la page de la SDL: <http://www.libsdl.org/index.php> . Allez sur "Download" et "SDL 1.2". Vous aurez alors 2 archives à télécharger:

La première se trouve dans "Runtime Libraries" et se nomme "SDL-1.2.15-win32.zip". A l'intérieur se trouve le fichier "SDL.dll" que vous devez placer dans le dossier de votre projet (à côté de votre fichier .cbp).

La seconde archive se trouve dans "Development Libraries" et se nomme "SDL-devel-1.2.15-mingw.tar.gz". Décompressez-la et copiez le dossier "SDL-1.2.15" dans le dossier de CodeBlocks, à côté du dossier MinGW.

Enfin, pour créer une application SDL, ouvrez Code:Blocks, créez un nouveau projet SDL (**SDL Project**), donnez un nom à votre programme puis lorsque Code:Blocks vous demande où se trouve le dossier SDL (**Please select SDL's location**), choisissez le dossier "SDL-1.2.15" que vous avez placé à côté du dossier MinGW (si par la suite une nouvelle fenêtre s'ouvre et vous demande de configurer les différents emplacements de vos headers et de vos lib, ne modifiez que la section "Base" en indiquant à nouveau le chemin du dossier "SDL-1.2.15").

De plus, comme CodeBlocks fournit un template pour créer un projet SDL, une fois cette dernière installée, vous n'aurez pas besoin de linker les lib, CodeBlocks le fera tout seul.

2/ Structures et Fonctions SDL pour débiter

SDL_Surface

est structure utilisée pour tout élément graphique, de l'écran aux images chargées. Du point de vue de SDL, l'écran est une surface presque comme une autre.

SDL_Rect

est une structure C qui définit un rectangle. Les champs de SDL_Rect sont :

```
int    x      Position  x du rectangle (le coin haut gauche)
int    y      Position  y du rectangle (le coin haut gauche)
int    w      Largeur du rectangle
int    h      Hauteur rectangle
```

Elle est utilisée pour définir ou placer les surfaces.

SDL_SetVideoMode

met en place un mode vidéo avec la largeur, la hauteur, la quantité de bit par pixel et d'autres paramètres vidéo par exemple la gestion de la mémoire.

```
SDL_Surface *fen;
```

```
fen = SDL_SetVideoMode(640, 480, 16, SDL_HWSURFACE);
```

```
SDL_SWSURFACE    Crée une Surface en mémoire centrale
```

```
SDL_HWSURFACE    Crée une Surface en mémoire Vidéo
```

```
SDL_DOUBLEBUF    Utilise le mode Double Buffer des cartes graphiques
```

SDL_LoadBMP

Charge une image au format bmp

```
SDL_Surface *ima1;
```

```
ima1 = SDL_LoadBMP("cb.bmp");
```

SDL_Event

Structure C pour enregistrer et gérer les événements Un événement est simplement quelque chose qui se passe. Cela peut être la touche qu'on presse, le mouvement de la souris, le redimensionnement de la fenêtre ou la fermeture de la fenêtre.

```
SDL_Event event;
```

SDL_MapRGB

On utilise cette fonction pour associer (mapper) un triplet RVB à une valeur de pixel opaque pour un format donné. `SDL_MapRGB(const SDL_PixelFormat* format, Uint8 R, Uint8 V, Uint8 B)`

SDL_FillRect

Cette fonction remplit un rectangle avec la couleur choisie

```
SDL_FillRect(fen, 0, SDL_MapRGB(fen->format, 255, 0, 0));
```

SDL_BlitSurface

Cette fonction plaque l'image ima1 dans la fenêtre fen.

```
SDL_BlitSurface(ima1, 0, fen, &pos1);
```