Tutoriel SDL

1 Installer la SDL en local

1. Télécharger la SDL version 2 :
   https://www.libsdl.org/download-2.0.php
2. Pour Unix, prendre le fichier SDL2-2.0.7.tar.gz, puis extraire le contenu de ce fichier avec la commande **tar xzvf SDL2-2.0.7.tar.gz** [Note : Ça doit vous rappeler le module de SE de la L1 !]
3. Procédure d'installation Dans un terminal, aller dans le répertoire ∼/**SDL2-2.0.7**
   (a) Exécuter : `./configure` --prefix=[install-dir]
      . `[install-dir]` est le répertoire d'installation
      . Ex : `./configure` --prefix=$HOME/SDL2
      . Tous les fichiers seront installés dans ce répertoire
   (b) Exécuter : `make`
      . Cela va compiler les fichiers sources
      . Cette étape peut prendre plusieurs minutes
   (c) Exécuter : `make` install
      . Cela va copier les binaires, bibliothèques de fonctions et fichier .h dans le répertoire d'installation `[install-dir]`
4. Dans `[install-dir]` nous obtenons (entre autre) :
   (a) Un répertoire `lib` contenant les bibliothèques de fonctions libSDL2.a (statique) et libSDL2.so (dynamique)
      . `libSDL2.so` nous intéresse : nous ferons une liaison dynamique avec la SDL.
   (b) Un répertoire `include` contenant les fichier `.h` et notamment `<SDL2/SDL.h>`

2 Configurer l'environnement de travail

Lors de la compilation de votre projet, nous allons lier dynamiquement la bibliothèque SDL. Elle ne sera donc pas incluse dans le fichier exécutable final. Lors de l'exécution, le système ira chercher la bibliothèque de fonctions dans les répertoires indiqués dans `LD_LIBRAY_PATH`. Il faut donc ajouter le chemin à cette variable d'environnement.

5. `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/SDL2/lib`
   - Pour rendre ce changement permanent, ajoutez la commande au fichier caché `$HOME/.bashrc`
   - Exécutez `source` ∼/.bashrc lorsque vous créez ou mettez à jour le fichier .bashrc (seulement la première fois)

Il faut ensuite créer un `makefile` pour votre projet afin de préciser les informations permettant au compilateur de trouver et include la SDL. Voici un makefile minimaliste qui permet de le faire :

```
CC=gcc
FLAGS=-Wall -g

SDL_DIR=${HOME}/SDL2
SDL_LIB_DIR=${SDL_DIR}/lib
SDL_INC_DIR=${SDL_DIR}/include

LIBS=-L${SDL_LIB_DIR} -lSDL2
INCS=-I${SDL_INC_DIR}
PROG=sdl_test

all: sdl_text

sdl_text: sdl_test.c
    ${CC} -o ${PROG} sdl_test.c ${LIBS} ${INCS} ${FLAGS}

clean:
    rm -f ${PROG}
    rm -f *.o
```

Explications :

- `-L` : permet de spécifier où trouver la bibliothèque `libSDL2.so`
- `-l` : permet de dire qu'il faut utiliser la bibliothèque `SDL2`
- `-I` : permet de spécificer où sont les fichiers `.h`

3 SDL2 Image et TTF

L'installation se fait de la même manière que pour al SDL :

. Télécharger la SDL2 Image : https://www.libsdl.org/projects/SDL_image

. Télécharger la SDL2 TTF : https://www.libsdl.org/projects/SDL_ttf

Puis installer de manière classique :

1. `./configure` --prefix=$HOME/SDL2
2. `make`
3. `make` install

Il ne reste plus qu'à modifier le Makefile pour inclure ces bibliothèque lors de la compilation : `LIBS=-L${SDL_LIB_DIR}` -lSDL2 -lSDL2_image -lSDL2_ttf

4 Utilisation / premier programme

```c
#include <stdio.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_image.h>

int main(int argc, char** argv)
{
    //Le pointeur vers la fenetre
        SDL_Window* pWindow = NULL;
        //Le pointeur vers la surface incluse dans la fenetre
    SDL_Surface *texte=NULL, *image=NULL;
        SDL_Renderer *renderer=NULL;
        SDL_Rect txtDestRect,imgDestRect;

        // Le pointeur vers notre police
        TTF_Font *police = NULL;
        // Une variable de couleur noire
        SDL_Color couleurNoire = {0, 0, 0};

    /* Initialisation simple */
    if (SDL_Init(SDL_INIT_VIDEO) != 0 ) {
        fprintf(stdout,"Echec de l'initialisation de la SDL (%s)\n",SDL_GetError());
        return -1;
    }

        /* Initialisation TTF */
        if(TTF_Init() == -1) {
                fprintf(stderr, "Erreur d'initialisation de TTF_Init : %s\n", TTF_GetError());
                exit(EXIT_FAILURE);
        }


        /* Creation de la fenetre */
        pWindow = SDL_CreateWindow("Hello World SDL2",SDL_WINDOWPOS_UNDEFINED,
                SDL_WINDOWPOS_UNDEFINED,
                640,
                480,
                SDL_WINDOW_SHOWN|SDL_WINDOW_RESIZABLE);

        if(!pWindow){
                fprintf(stderr, "Erreur a la creation de la fenetre : %s\n", SDL_GetError());
                exit(EXIT_FAILURE);
        }


        renderer = SDL_CreateRenderer(pWindow, -1, SDL_RENDERER_ACCELERATED);
        if(renderer == NULL){
                fprintf(stderr, "Erreur a la creation du renderer\n");
                exit(EXIT_FAILURE);
        }


        if( (police = TTF_OpenFont("ChowFun.ttf", 20)) == NULL){
                fprintf(stderr, "erreur chargement font\n");
                exit(EXIT_FAILURE);
        }
        texte = TTF_RenderUTF8_Blended(police, "Vive la programmation !", couleurNoire);
        if(!texte){
                fprintf(stderr, "Erreur a la creation du texte : %s\n", SDL_GetError());
                exit(EXIT_FAILURE);
        }

        SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255);
        SDL_Texture *texte_tex = SDL_CreateTextureFromSurface(renderer, texte);
        if(!texte_tex){
                fprintf(stderr, "Erreur a la creation du rendu du texte : %s\n", SDL_GetError());
                exit(EXIT_FAILURE);
        }
        SDL_FreeSurface(texte); /* on a la texture, plus besoin du texte */
        /* Position ou sera mis le texte dans la fenetre */
        txtDestRect.x = txtDestRect.y = 10;
        SDL_QueryTexture(texte_tex, NULL, NULL, &(txtDestRect.w), &(txtDestRect.h));

        // load sample.png into image
        SDL_RWops *rwop=SDL_RWFromFile("affiche2018.png", "rb");
        image=IMG_LoadPNG_RW(rwop);
        if(!image) {
             printf("IMG_LoadPNG_RW: %s\n", IMG_GetError());
        }
```

```
        SDL_Texture *image_tex = SDL_CreateTextureFromSurface(renderer, image);
        if(!image_tex){
                fprintf(stderr, "Erreur a la creation du rendu de l'image : %s\n", SDL_GetError());
                exit(EXIT_FAILURE);
        }
        SDL_FreeSurface(image); /* on a la texture, plus besoin de l'image */



        if( pWindow )
        {
                int running = 1;
                while(running) {
                        SDL_Event e;
                        while(SDL_PollEvent(&e)) {
                                switch(e.type) {
                                        case SDL_QUIT: running = 0;
                                        break;
                                        case SDL_WINDOWEVENT:
                                                switch(e.window.event){
                                                        case SDL_WINDOWEVENT_EXPOSED:
                                                        case SDL_WINDOWEVENT_SIZE_CHANGED:
                                                        case SDL_WINDOWEVENT_RESIZED:
                                                        case SDL_WINDOWEVENT_SHOWN:
                                /* Le fond de la fenetre sera blanc */
                                SDL_SetRenderDrawColor(renderer, 255, 255, 255, 255);
                                SDL_RenderClear(renderer);

                                /* Ajout du texte en noir */
                                SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255);
                                SDL_RenderCopy(renderer, texte_tex, NULL, &txtDestRect);

                                /* Ajout de la seconde image a une certaine position */
                                imgDestRect.x = imgDestRect.y = 50;
                                SDL_QueryTexture(image_tex, NULL, NULL, &(imgDestRect.w),
                                    &(imgDestRect.h));
                                SDL_RenderCopy(renderer, image_tex, NULL, &imgDestRect);

                                /* Ajout de la seconde image a une autre position */
                                imgDestRect.x = 250;
                                SDL_RenderCopy(renderer, image_tex, NULL, &imgDestRect);

                                /* On fait le rendu ! */
                                SDL_RenderPresent(renderer);

                                                        break;
                                                }
                                        break;
                                }
                        }
                }
        } else {
                fprintf(stderr,"Erreur de creation de la fenetre: %s\n",SDL_GetError());
        }

        //Destruction de la fenetre
        SDL_DestroyWindow(pWindow);

        TTF_CloseFont(police); /* Doit etre avant TTF_Quit() */
        TTF_Quit();
    SDL_Quit();

    return 0;
}
```

5 ressources

De nombreux tutoriels en ligne sont disponibles. En voici quelques-uns :

1. developpez.com
2. Blog de Will Usher
3. Openclassroom