



Librairie SDL

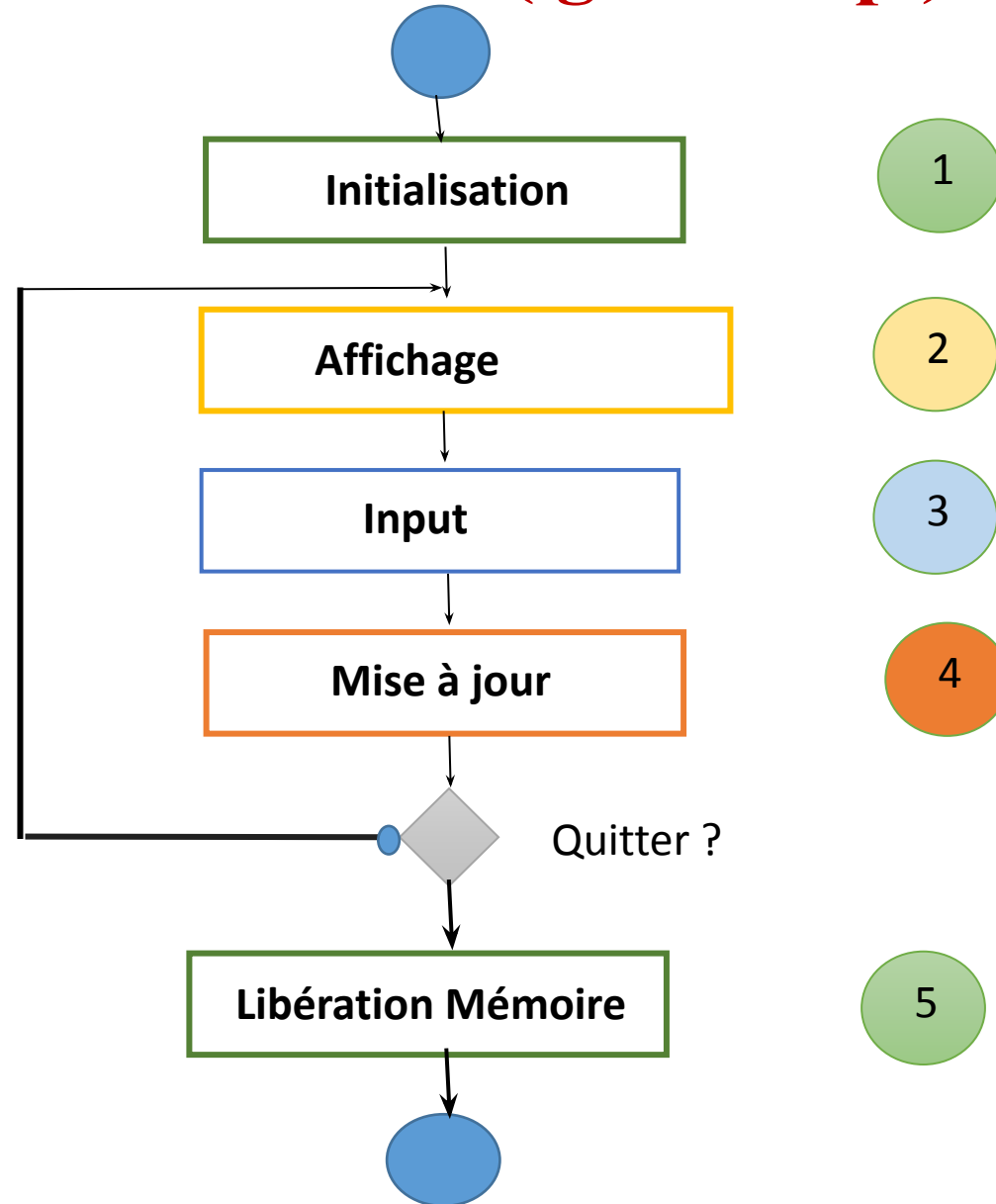
SDL: Simple DirectMedia Layer

- **Bibliothèque tierce** : non installée par défaut.
- **Bibliothèque écrite en C**
- **Bibliothèque libre et gratuite**
- **Bibliothèques multi-plates-formes**
- **Pour l'installer: tapez la commande suivante sur vos terminaux:**

```
sudo apt-get install libsdl1.2-dev libsdl-image1.2-dev libsdl-mixer1.2-dev libsdl-ttf2.0-dev libpulse-dev  
libxt-dev
```

- **Après installation, les bibliothèques SDL sont ajoutées dans le chemin suivant:**
[/usr/include/SDL/](#)

Boucle de Jeu (gameloop)



Initialisation de la SDL

SDL est composée de sous systèmes : vidéo, audio, timing... Vous pouvez initialiser quelques composantes en précisant leurs noms ou bien initialiser toutes les composantes en utilisant le flag "EVERYTHING"

--> **L'Initialisation** est assurée grâce à la fonction **SDL_Init** qui retourne 0 si l'initialisation s'est déroulée avec succès.

Exemple :

avant d'afficher
une image

avant de récupérer le nb de
secondes depuis l'initialisation
de SDL

avant de jouer
un son

```
if(SDL_Init ( SDL_INIT_VIDEO | SDL_INIT_TIMER | SDL_INIT_AUDIO ) !=0)
{
    printf("Erreur : %s \n", SDL_GetError() );
    return 1 ;
}
```

SDL_Quit(): permet de libérer les ressources utilisées par les sous systèmes SDL déjà initialisés.

Création d'une fenêtre

SDL_SetVideoMode (largeur , hauteur, nombre de couleurs affichables, flags);
permet de créer une fenêtre dont la largeur , l'hauteur et la résolution (en pixels) sont passés en paramètres.

Exemple :

`SDL_SetVideoMode (400,400,32, SDL_HWSURFACE | SDL_DOUBLEBUF) ;`

Flags



8 bits / pixel



32 bits / pixel

Exemples de Résolution

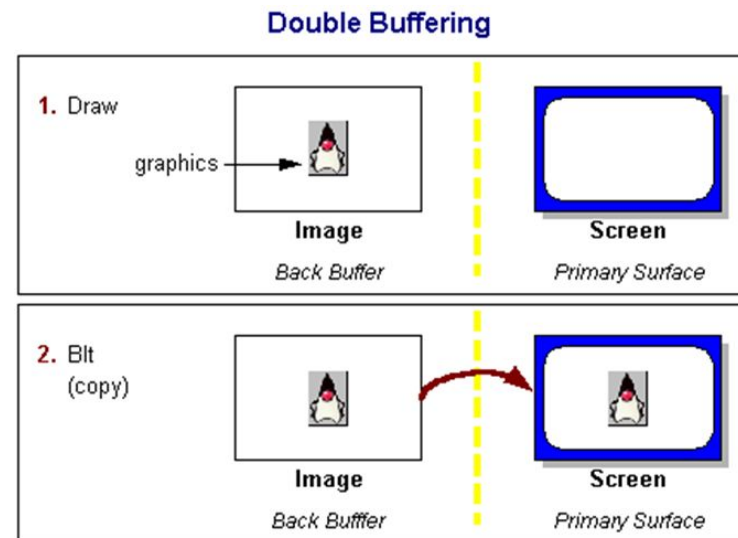
Les flags de SDL_SetVideoMode

SDL_SWSURFACE : Sauvegarde des données dans la RAM

SDL_HWSURFACE : Sauvegarde des données dans la mémoire vidéo de la carte graphique (VRAM)

SDL_RESIZABLE : Agrandir / rétrécir une fenêtre

SDL_DOUBLEBUF : technique permettant un mouvement fluide des objets



Afficher une image

Si vous manipulez des images .bmp, vous pouvez vous contenter de la bib **SDL.h**

Sinon, il faut inclure la bibliothèque: **SDL_image.h**

```
#include <SDL/SDL_image.h>
```

❖ Déclaration d'une surface ²²

```
SDL_Surface *image;
```

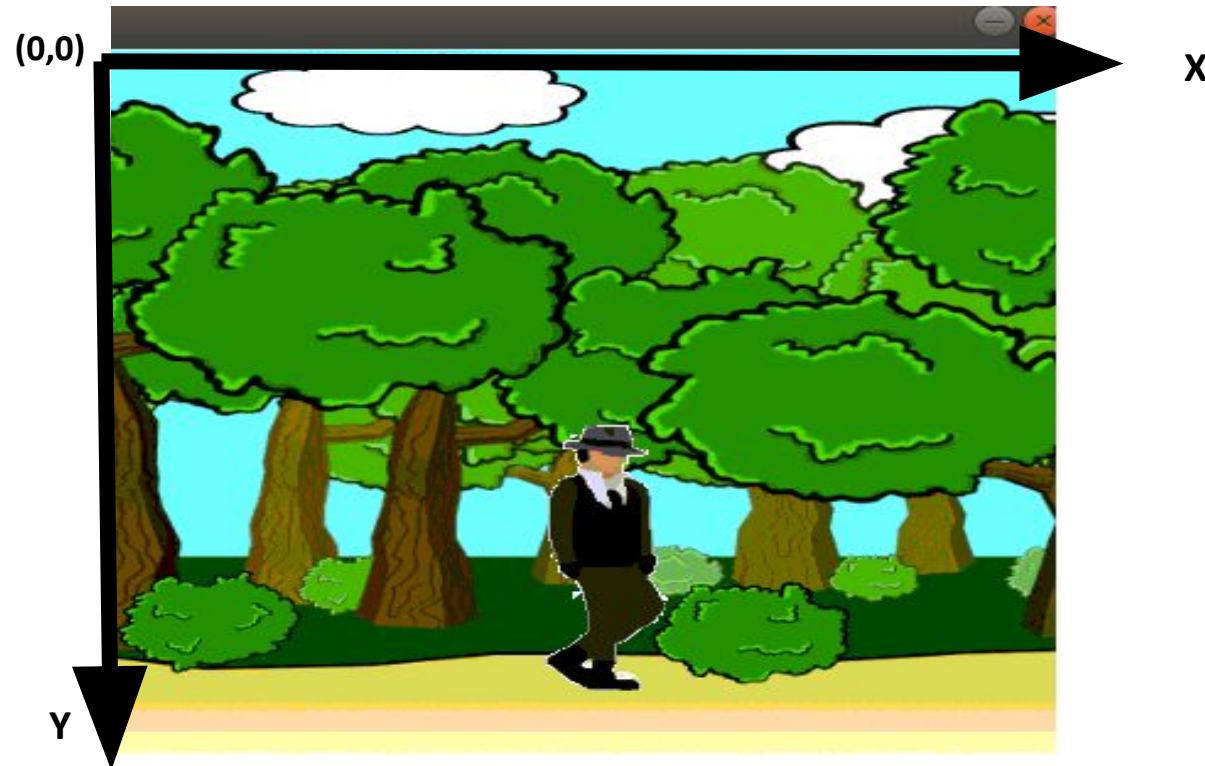
❖ Chargement d'image dans une surface

```
image=IMG_Load(nomFichier); // pour n'importe quel type d'images
```

```
image=SDL_LoadBMP (nomFichier); // seulement pour les images bitmap
```

Echelle / Repère dans une Surface SDL

Que ce soit la surface est un écran ou une image un repère SDL est représentée ainsi:



Le type **SDL_Rect**

Le type **SDL_Rect** définit une zone rectangulaire. Il est utilisé par `SDL_BlitSurface ()` pour définir les régions de collage.

Il s'agit d'une structure prédéfinie dans la bib `SDL.h`. Cette structure est composée des champs suivants:

- x: l'abscisse
- y: l'ordonnée
- w: la largeur
- h: l'hauteur

Afficher une image

❖ Pour coller une image sur un écran on utilise la fonction :

SDL_BlitSurface(*La surface source à copier* `SDL_Surface* src`, *La région source* `const SDL_Rect* srcrect`, *La surface destination: Ecran* `SDL_Surface* dst`, *L'emplacement destination* `SDL_Rect* dstrect`);

srcrect: la valeur de ce champ vaut:

- NULL: si vous souhaitez afficher la totalité de l'image
- adresse d'une variable de type `SDL_Rect`: si vous souhaitez afficher une partie de l'image.

dstrect: la valeur de ce champ vaut l'adresse d'une position pos déclarée et initialisée précédemment.

❖ Mise à jour de l'écran:

La mise à jour de l'écran consiste à rafraîchir l'écran après avoir effectué plusieurs collages

SDL_Flip(`SDL_Surface *screen`);

Exemple: afficher l'image de background et la deuxième image de Neymar

```
#ifndef Image_H
#define Image_H
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#define SCREEN_W 800
#define SCREEN_H 400
struct Image{
    SDL_Rect pos1;
    SDL_Rect pos2;
    SDL_Surface * img;
};
typedef struct Image Image;

void initPlayer(Image *A);
void initBackground(Image *Backg);
void liberer(Image A);
void afficher(Image p,SDL_Surface *screen);
#endif
```

position de l'image par rapport à l'écran

quelle partie de l'image doit être affichée

image.h

image.c

Exemple: afficher l'image de background et le deuxième image de Neymar

```
#include "texte.h"
#include "Image.h"

int main(int argc, char** argv)
{
    // initialisation texte, video et son
    TTF_Init();
    SDL_Init(SDL_INIT_VIDEO);

    //Déclaration des variables
    SDL_Surface *screen;
    int done = 0;
    Image Hazard, Backg;
    SDL_Event event;
    Text t;

    atexit(SDL_Quit);
    // creation d'une fenetre
    screen = SDL_SetVideoMode(SCREEN_W, SCREEN_H, 32, SDL_HWSURFACE | SDL_SRCALPHA);
    if ( !screen ) {
        printf("Unable to set 600x300 video: %s\n", SDL_GetError());
        return 1;
    }

    //initialisation des variables
    initBackground(&Backg);
    initPlayer(&Hazard);
    initText(&t);

    // Game loop
    while(!done) {
        //Display
        afficher(Backg, screen);
        afficher(Hazard, screen);
        displayText(t, screen);
        SDL_Flip(screen); //Flip the backbuffer to the primary Hardware Video Memory
        //Input
        SDL_PollEvent(&event); //difference avec wait event ca gene quand il ya update independant de event
        switch (event.type) {
            // exit if the window is closed
            case SDL_QUIT:
                done = 1;
            break;
        }
    }

    //cleanup
    freeText(t);
    TTF_CloseFont (t.font);
    TTF_Quit();
    return 0;
}
```

main.c

Exemple: afficher l'image de background et la deuxième image de Neymar

```
#ifndef Image_H
#define Image_H
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#define SCREEN_W 800
#define SCREEN_H 400
struct Image{
    SDL_Rect pos1;
    SDL_Rect pos2;
    SDL_Surface * img;
};
typedef struct Image Image;

void initPlayer(Image *A);
void initBackground(Image *Backg);
void liberer(Image A);
void afficher(Image p,SDL_Surface *screen);
#endif
```

position de l'image par rapport à l'écran

quelle partie de l'image doit être affichée

```
#include <SDL/SDL.h>
#include "Image.h"
#include <SDL/SDL_image.h>

void initPlayer(Image *A)
{
    A->img = IMG_Load("../images/player.png");
    if (A->img== NULL) {
        return ;
    }
    A->pos1.x=0;
    A->pos1.y=100;
    A->pos1.w=100;
    A->pos1.h=A->img->h;

    A->pos2.x=193;
    A->pos2.y=0;
    A->pos2.w=100;
    A->pos2.h=A->img->h;;
}

void initBackground(Image *Backg)
{
    Backg->img= SDL_LoadBMP("../images/background.bmp");
    if (Backg->img == NULL) {
        printf("Unable to load bitmap: %s\n", SDL_GetError());
        return;
    }

    Backg->pos1.x=0;
    Backg->pos1.y=0;
    Backg->pos2.x=0;
    Backg->pos2.y=0;
    Backg->pos2.w=(SCREEN_W);
    Backg->pos2.h=SCREEN_H;
}

void liberer(Image A)
{
    SDL_FreeSurface(A.img);
}

void afficher(Image p,SDL_Surface *screen)
{
    SDL_BlitSurface(p.img,&p.pos2,screen,&p.pos1);
}
```

image.h

image.c

Exemple: afficher l'image de background et la deuxième image de Neymar

```
#include "texte.h"
#include "image.h"
int main ( int argc, char** argv )
{
    // initialisation texte, video et son
    TTF_Init();
    SDL_Init( SDL_INIT_VIDEO );
    //Déclaration des variables
    SDL_Window *window;
    SDL_Renderer *renderer;
    int done = 0;
    Image Backg, Hazar;
    SDL_Event event;
    Text t;
    atexit(SDL_Quit);
    // creation d'une fenetre
    screen = SDL_SetVideoMode(SCREEN_W, SCREEN_H, 32,SDL_HWSURFACE | SDL_SRCALPHA);
    if ( !screen ) {
        printf("Unable to set 600x300 video: %s\n", SDL_GetError());
        return 1;
    }
    //Initialisation des variables
    initBackground(&Backg);
    initPlayer(&Hazar);
    initText(&t);
    // Game loop
    while ( !done ) {
        //Display
        afficher(Backg,screen);
        afficher(Hazar,screen);
        displayText(t, screen);
        SDL_Flip(screen); //Flip the backbuffer to the primary Hardware Video Memory
        //Input
        SDL_PollEvent(&event); //difference avec wait event ca gene quand il ya update independant de event
        switch (event.type) {
            // exit if the window is closed
            case SDL_QUIT:
                done = 1;
                break;
        }

        liberer(Backg);
        freeText(t);
        TTF_CloseFont (t.font);
        TTF_Quit();
        return 0;
    }
}
```

main.c

Afficher un texte

Un texte est une surface conçue à partir d'une chaîne de caractère, une couleur, une police et une taille.

Pour créer un texte, il faut inclure la bibliothèque `<SDL/SDL_ttf.h>` et suivre les étapes suivantes:

1. **Initialisation de SDL_ttf** : `TTF_Init();`

2. **Chargement d'une police et initialisation de la taille**

```
TTF_Font *police = TTF_OpenFont( fichier de police, Taille );
```

Exemple :

```
police = TTF_OpenFont ("times.ttf", 65);
```

1. **initialisation de la couleur**

`SDL_Color textColor=(R,G,B);` la valeur de R, G, B doit être entre 0 et 255

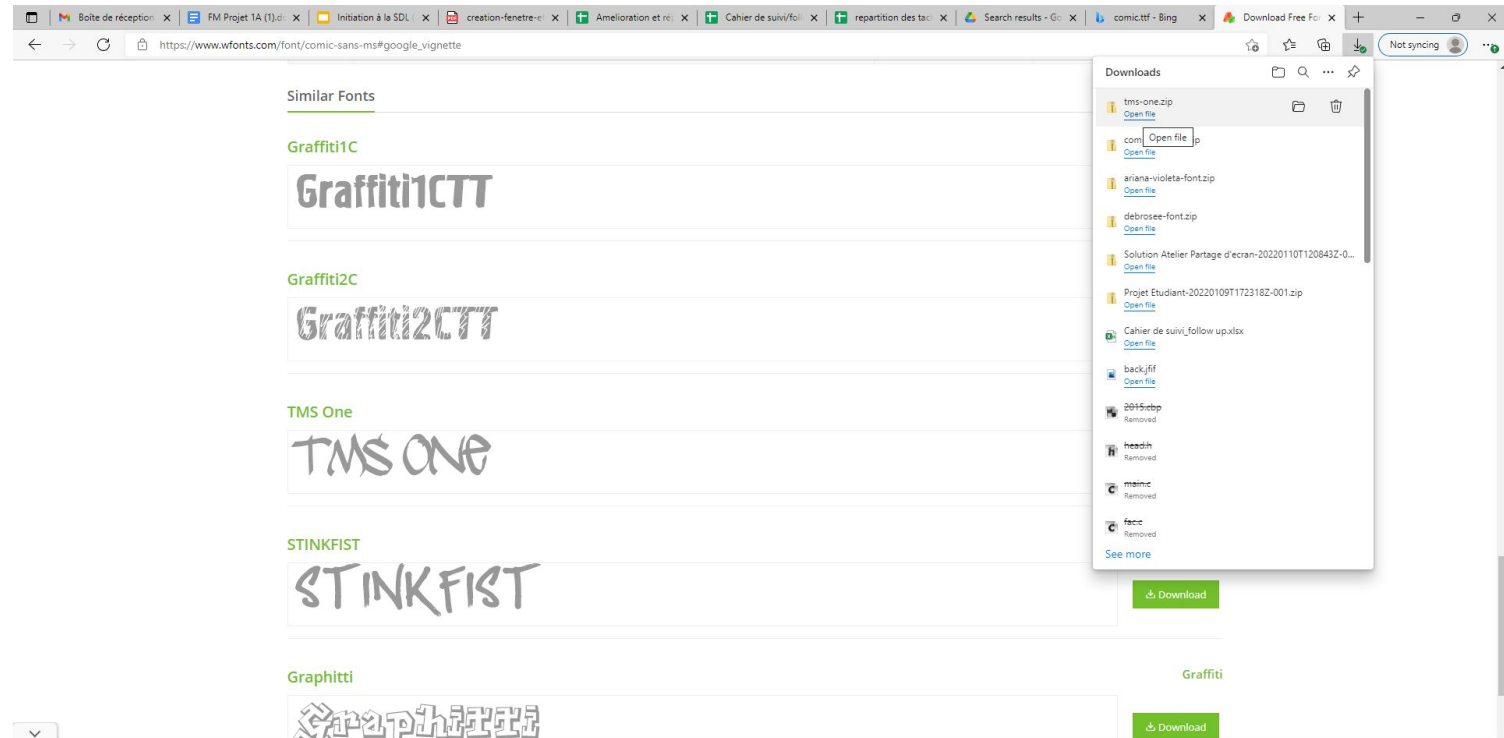
Pour plus d'informations sur les codes couleurs consulter [cette palette](#).

Afficher un texte: exemple de codes couleur

Color	Color name	(R,G,B)
	Black	(0,0,0)
	White	(255,255,255)
	Red	(255,0,0)
	Lime	(0,255,0)
	Blue	(0,0,255)
	Yellow	(255,255,0)
	Cyan	(0,255,255)
	Magenta	(255,0,255)
	Silver	(192,192,192)
	Gray	(128,128,128)
	Maroon	(128,0,0)
	Olive	(128,128,0)
	Green	(0,128,0)
	Purple	(128,0,128)
	Teal	(0,128,128)
	Navy	(0,0,128)

Afficher un texte: Les Polices

- Sur Ubuntu, vous disposez déjà de plusieurs polices (fichiers avec extension .ttf) sous ce répertoire: */usr/share/fonts/truetype*
- Si vous souhaitez utiliser une autre police, vous pouvez la télécharger par exemple de [ce site](#).



Afficher un texte

3. Les méthodes d'écriture:

TTF_RenderText_Solid; si vous avez un texte qui change souvent.

TTF_RenderText_Shaded; si votre texte ne change pas très souvent et que vous voulez blitter votre texte sur un fond uni.

TTF_RenderText_Blended ; si votre texte ne change pas très souvent mais que vous voulez blitter sur un fond non uni (comme une image).

Exemple d'écriture de texte en Blended:

```
SDL_Surface *texte = NULL;
SDL_Color couleurNoir={0, 0, 0};
texte= TTF_RenderText_Blended (police , " Message " couleurNoir );
```

Chargement de la police Le message à afficher Couleur du message

4. Bliter la surface: **SDL_BlitSurface (texte, NULL, screen, &positionEcran);**

5. Fermer la police : **TTF_CloseFont (police);**

6. Arrêt de SDL_ttf : **TTF_Quit();**

Exemple: afficher le texte Neymar en jaune à la pos(300, 50)

```
#ifndef Texte_H
#define Texte_H
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
typedef struct {
    SDL_Rect position;
    TTF_Font *font;
    SDL_Surface * surfaceTexte;
    SDL_Color textColor;
    char texte [50];
} Text;

void initText(Text *t);
void freeText(Text A);
void displayText(Text t,SDL_Surface *screen);

#endif
```

texte.h

```
#include <SDL/SDL.h>
#include "texte.h"
#include <SDL/SDL_image.h>
// liberer memoire a chaque fois on quitte un stage

void initText(Text *A)
{
    A->position.x=300;
    A->position.y=50;
    // couleur jaune
    A->textColor.r=255;
    A->textColor.g=255;
    A->textColor.b=0;

    A->font = TTF_OpenFont( "arial.ttf", 70 );
}

void freeText(Text A)
{
    SDL_FreeSurface(A.surfaceTexte);
}

void displayText(Text t,SDL_Surface *screen)
{
    t.surfaceTexte = TTF_RenderText_Solid(t.font, "Neymar", t.textColor );
    SDL_BlitSurface(t.surfaceTexte, NULL, screen,&t.position);
}
```

texte.c

Exemple: afficher le texte Nevmar en jaune à la pos(300, 50)

```
#include "texte.h"
#include "Image.h"
int main ( int argc, char** argv )
{
    // initialisation texte, video et son
    TTF_Init();
    SDL_Init( SDL_INIT_VIDEO );
    //Déclaration des variables
    SDL_Surface *screen;
    int done = 0;
    Image Hazar, Backg;
    SDL_Event event;
    Text t;
    atexit(SDL_Quit);
    // création d'une fenetre
    screen = SDL_SetVideoMode(SCREEN_W, SCREEN_H, 32,SDL_HWSURFACE | SDL_SRCALPHA);
    if ( !screen ) {
        printf("Unable to set 600x300 video: %s\n", SDL_GetError());
        return 1;
    }
    //Initialisation des variables
    initBackground(&Backg);
    initPlayer(&Hazar);
    initText(&t);
    // Game loop
    while(!done) {
        //Display
        afficher(Backg,screen);
        afficher(Hazar,screen);
        displayText(t, screen);
        SDL_Flip(screen); //Flip the backbuffer to the primary Hardware Video Memory
        //Input
        SDL_PollEvent(&event); //difference avec wait event ca gene quand il ya update independant de event
        switch (event.type) {
            // exit if the window is closed
            case SDL_QUIT:
                done = 1;
                break;
        }
    }
    // free Surfaces
    liberer(Hazar);
    liberer(Backg);
    freeText(t);
    TTF_CloseFont (t.font);
    TTF_Quit();
    return 0;
}
```

main.c

Jouer un son continu

SDL_mixer:

- Bibliothèque de mixage de sons.
- Permet la gestion de l'audio.

1. Initialiser les fonctions audio de SDL_mixer:

int **Mix_OpenAudio**(int *frequency*, Uint16 *format*, int *channels*, int *chunksiz*);

Fréquence d'échantillonnage (Hz) Format d'échantillonnage Nombre de canaux sonores Octets/échantillon

2. Chargement de la musique:

```
Mix_Music *musique;
musique = Mix_LoadMUS("musique.mp3");
```

3. Jouer la musique

```
int Mix_PlayMusic(Mix_Music *music, int loops);
```

Nb de répétition (-1 infini)

4. Libération de la musique

```
Mix_FreeMusic(musique);
```

Jouer un son continu

Exemple:

```
#include <SDL/SDL_mixer.h>

if(Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, MIX_DEFAULT_CHANNELS, 1024) == -1)
//API Mixer Initialization
{
    printf("%s", Mix_GetError());
}
Mix_Music *music; //Construct Mix_Music pointer
music = Mix_LoadMUS("music.mp3"); //load the music
Mix_PlayMusic(music, -1); //play music
.....

Mix_FreeMusic(music);
```

Jouer un son bref

1. Chargement des effets sonores:

```
Mix_Chunk * son;  
son= Mix_LoadWAV( "effet.wav" );
```

2. Jouer l'effet sonore

```
Mix_PlayChannel( (-1), (son), (0) );
```

Le canal
Le Mix_Chunk à jouer
Nb de répétition du son

3. Libération des effets sonores

```
Mix_FreeChunk( son );
```

Les événements en SDL

Un évènement: un « signal » envoyé par un périphérique à votre application.

- Appuie sur une touche du clavier
- Clic avec la souris
- Bouger la souris
- Réduire/ fermer la fenêtre

Les événements en SDL

Pour lire un événement, vous devez:

1. déclarer une variable de type `SDL_Event`
2. Lire l'événement avec une de ces de fonctions:
 - `SDL_Pollevnt`: lit l'évènement sans bloquer la boucle du jeu
 - `SDL_waitEvent`: reste en attente jusqu'à écouter un évènement

Remarque: si vous avez des traitements automatiques qui ne dépendent pas de l'utilisateur tels que une animation de background, un déplacement d'ennemis..., vous avez intérêt à utiliser la première fonction: `SDL_PollEvent`

Les événements en SDL

- Soit le code suivant permettant de lire un événement:

SDL_Event event;

SDL_PollEvent(&event); // a appeler dans la boucle du jeu

- Pour détecter un Appuie sur une touche clavier, il suffit de comparer le type de l'événement avec :

if(event.type==SDL_KEYDOWN)

- Pour savoir qu'elle touche a été appuyée, vous devez faire cette comparaison:

if(event.key.keysym.sym == nom_de_la_touche)

Exemple: *if(event.key.keysym.sym ==SDLK_ESCAPE)*

SDL KEY CODE

Values

<i>Key Name</i>	<i>SDL_Keycode Value</i>
"0"	SDLK_0
"1"	SDLK_1
"2"	SDLK_2
"3"	SDLK_3
"4"	SDLK_4
"5"	SDLK_5
"6"	SDLK_6
"7"	SDLK_7
"8"	SDLK_8
"9"	SDLK_9
"A"	SDLK_a
"Space" (the Space Bar key(s))	SDLK_SPACE

Les événements en SDL

- Pour détecter un mouvement de la souris, il suffit de comparer le type de l'événement avec :

if(event.type==SDL_MOUSEMOTION)

- Pour détecter un clic de la souris, il suffit de comparer le type de l'événement avec :

if(event.type==SDL_MOUSEBUTTONDOWN) //ou SDL_MOUSEBUTTONUP

- Pour savoir où le mouvement/clic de la souris a eu lieu, il faut tester si la position de la souris est au-dessus de la surface désirée de coordonnées (X,Y,W,H):

*if(event.motion.x >= X && event.motion.x <= X+W &&
event.motion.y >= Y && event.motion.y <= Y+H)*

Les événements en SDL

Exemple:

```
SDL_Event event;
int done=1;
while (done)
{
```

Déclarer une variable de type SDL_Event

```
    while (SDL_PollEvent(&event))
    { // check for messages
        switch (event.type)
        {
            // exit if the window is closed
            case SDL_QUIT:
                done = 0;
                break;
            // check for keypresses
            case SDL_KEYDOWN:
            {
                // exit if ESCAPE is pressed
                if (event.key.keysym.sym == SDLK_ESCAPE)
                    done = 0;

                break;
            }
        } // end switch
    } // end of message processing
```

Lecture des événements
Renvoie 1 s'il y a un événement
0 sinon

Travail à faire

Implémentation de la partie 1 du cahier des charges du Menu.

Remarque: Lors de la compilation d'un code faisant appel aux fonctions de la bibliothèque SDL, vous devez ajouter les options suivantes dans l'étape d'édition des liens:

-lSDL

-lSDL_ttf

-lSDL_image

-lSDL_mixer

Exemple:

```
prog:main.o Image.o texte.o
    gcc main.o Image.o texte.o -o prog -lSDL -lSDL_ttf -g -lSDL_image -lSDL_mixer
main.o:main.c
    gcc -c main.c -g
Image.o:Image.c
    gcc -c Image.c -g
texte.o:texte.c
    gcc -c texte.c -g
```

Références

- Documentation SDL: <https://www.libsdl.org>, consulté le 26/01/2022