Chapitre III : SDL_TTF

par Loka

Date de publication : 30/03/2006

Dernière mise à jour : 17/04/2006

III - SDL_TTF

III - SDL_TTF

SDL_TTF est une extension de SDL qui permet l'emploie des polices true type (*.ttf) et leur affichage.

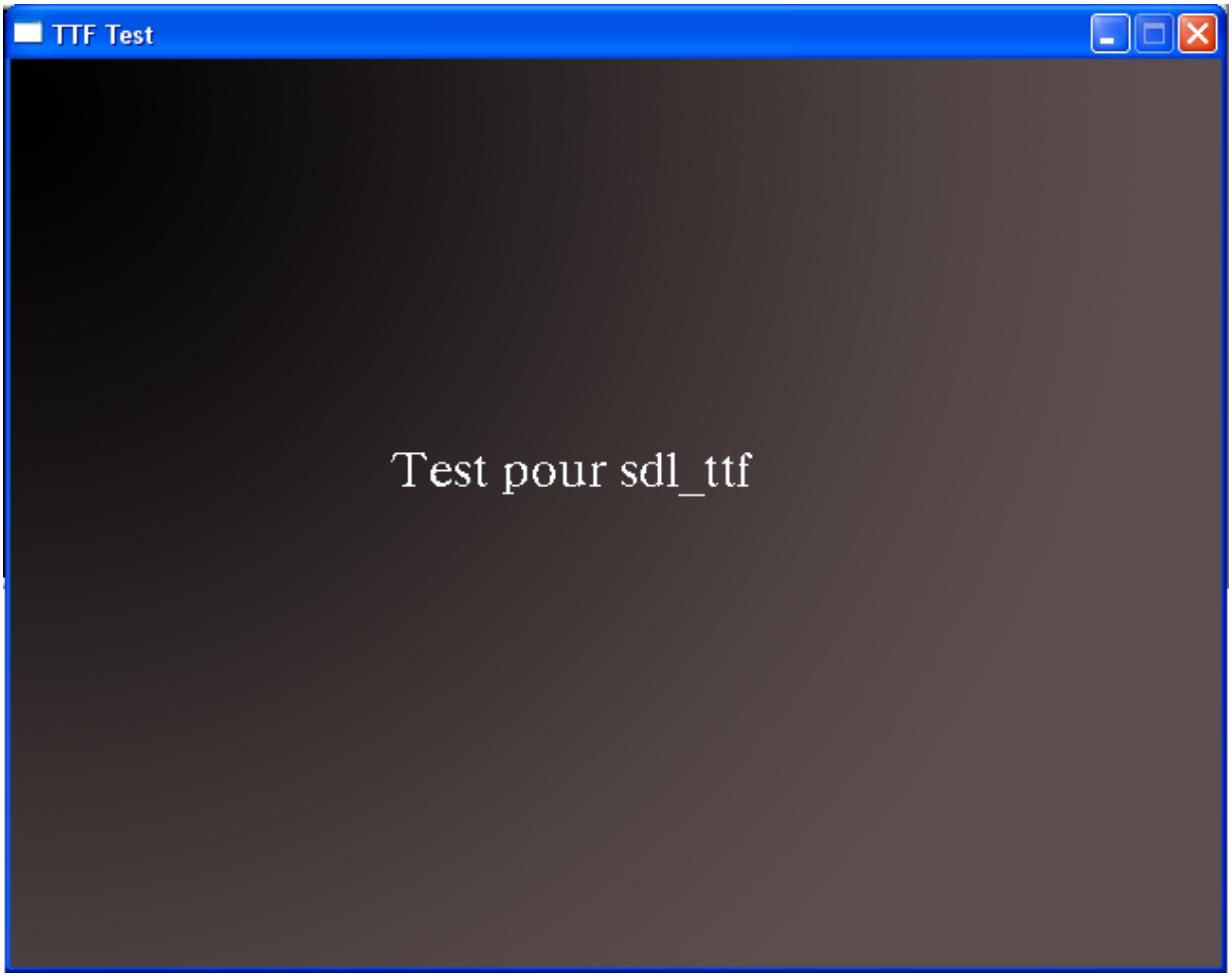
Tout d'abord il vous faudra installer cette extension, il vous suffit pour cela de le télécharger ici.

Si des demandes se font pour l'installation de cette extension (voir des autres extensions telles que sdl_image, sdl_net, etc...), je ferais un petit tutoriel afin de vous expliquez comment on fait sur les differents IDE.

Bien passons aux choses sérieuses.

SDL_TTF est assez simple d'utilisation, dans ce tutoriel, nous allons voir les bases d'utilisation de cette extension.

Voici le résultat de ce que nous allons chercher à obtenir :



Comme dans tout programme, il faut d'abord déclarer voire initialiser nos variables.

Déclaration et initialisation
```
//Les surfaces
SDL_Surface *background = NULL;
SDL_Surface *message = NULL;
SDL_Surface *screen = NULL;

//La structure d'événement
SDL_Event event;

//Le Font qu'on va utiliser
TTF_Font *font;

//La couleur du Font
SDL_Color textColor = { 255, 255, 255 };
```

Nous avons donc comme avant nos surfaces background et screen ainsi que notre structure d'événement event.

Apparaît en plus une surface *message* qui sera la surface qui contiendra notre texte.

Il y a aussi de nouvelles données de type **TTF_Font** qui sera le font (police de caractère) qu'on va utiliser et SDL_Color qui sera la couleur du texte.

Dans ce cas là, le texte sera blanc.

Nous allons maintenant reprendre notre fonction init() afin d'initialiser **SDL_TTF**

Initialisation
```
bool init() {
        //Initialisation de tous les sous-systèmes de SDL
        if( SDL_Init( SDL_INIT_EVERYTHING ) == -1 ) {
                return false;
        }

        //Mise en place de l'écran
        screen = SDL_SetVideoMode( SCREEN_WIDTH, SCREEN_HEIGHT, SCREEN_BPP, SDL_SWSURFACE );

        //S'il y a une erreur dans la mise en place de l'écran
        if( screen == NULL ) {
                return false;
        }

        //Initialisation de SDL_TTF
        if( TTF_Init() == -1 ) {
                return false;
        }

        //Mise en place de la barre caption
        SDL_WM_SetCaption( "TTF Test", NULL );

        //Si tout s'est bien passé
        return true;
}
```

SDL_TTF est initialisé en faisant appel à **TTF_init()**.

TTF_init() retourne -1 quand il y a une erreur.

TTF_init() doit être appelé avant d'utiliser n'importe quelle fonction de SDL_TTF.

Nous allons maintenant voir ce que nous avons à changer dans notre fonction de chargement load_files() afin de

charger la police de caractère que nous allons utiliser.

load_files
```
bool load_files() {
        //Chargement de l'image de fond
        background = load_image( "background.png" );

        //Ouverture du Font
        font = TTF_OpenFont( "CaslonBold.ttf", 28 );

        //S'il y a un problème au chargement du fond
        if( background == NULL ) {
                return false;
        }

        //S'il y a une erreur dans le chargement du Font
        if( font == NULL ) {
                return false;
        }

        //Si tout s'est bien chargé
        return true;
}
```

Le premier argument de **TTF_OpenFont** est le nom du fichier *.ttf que nous avons besoin d'ouvrir, le second argument est la taille à laquelle vous souhaitez initialiser le Font quand vous l'ouvrez.

Quand il y a une erreur dans le chargement du Font, TTF_OpenFont() retourne NULL.

Après avoir initialisé SDL_TTF et ouvert le font dont on a besoin, il nous reste alors à le "poser" sur l'écran.

Pour cela, nous allons utiliser la façon la plus rapide de rendre le texte en utilisant **TTF_RenderText_Solid()**.

TTF_RenderText_Solid
```
//Mise en place du texte sur la surface message
message = TTF_RenderText_Solid( font, "Test pour sdl_ttf", textColor );

//S'il y a une erreur dans la mise en place du texte
if( message == NULL ) {
        return 1;
}

//Application des images sur l'écran
apply_surface( 0, 0, background, screen );
apply_surface( 0, 200, message, screen );

//Mise à jour de l'écran
if( SDL_Flip( screen ) == -1 ) {
        return 1;
}
```

TTF_RenderText_Solid() prend le font dans le premier argument, et crée une surface avec le texte du second argument avec la couleur du troisième argument.

TTF_RenderText_Solid() retourne NULL quand il y a une erreur qui s'est produite.

Il y a bien évidemment bien d'autres façons de mettre en place le texte, vous pouvez allez chercher ça dans la documentation de SDL_TTF

Voilà, notre programme est maintenant censé afficher ce que nous souhaitons, il nous reste plus qu'à le quitter proprement.

Netoyage
```
   void clean_up() { //Libération des surfaces
           SDL_FreeSurface( background );
           SDL_FreeSurface( message );

           //Fermeture des Fonts qu'on a utilisé
           TTF_CloseFont( font );

           //On quitte SDL_ttf
           TTF_Quit();

           //On quitte SDL
           SDL_Quit();
   }
```

Voici notre fonction clean_up() retravaillé afin de permettre le nettoyage de sdl_ttf.

Premièrement nous libérons les surfaces utilisées, donc la surface message générée aussi.

Nous fermons aussi le Font en utilisant TTF_CloseFont() et quittons SDL_TTF grâce à TTF_Quit().

Il nous reste plus qu'à quitter SDL comme d'habitude.

Version pdf

Télécharger les sources

<= Retour au chapitre II : Premières applications

-=Index Tutos SDL=-

=> Accéder au chapitre IV : Gestion des événements