Guide SDL

Annexe

Par

KARL MONGOSSO
ANTOINE DAFLON

Mathématiques Appliquées et Informatique Numérique
SORBONNE UNIVERSITÉ

JANVIER 2018

Les Structures

- SDL_Texture : Structure permettant de gérer les données liées au pixel
- SDL_Window : Structure permettant de gérer la fenêtre graphique
- SDL_Renderer : Structure contenant l'état du rendu (renderer)
- SDL_Event : Union de structure contenant les structures des différents évènements
- SDL_Rect : Structure contenant la définition d'un rectangle avec l'origine en haut à gauche du rectangle
- SDL_Surface : Structure contenant un ensemble de pixel

Les Fonctions

- SDL_Init(Uint32 flags) : Initialise la librairie SDL
- SDL_Window* SDL_CreateWindow(const char* title, int x, int y, int w, int h, Uint32 flags) : Crée la fenêtre graphique à la position et aux dimensions spécifiées
- SDL_Renderer* SDL_CreateRenderer(SDL_Window* window, int index, Uint32 flags) : Crée un rendu 2D ajusté à la fenêtre graphique
- SDL_Texture* SDL_CreateTextureFromSurface(SDL_Renderer* renderer, SDL_Surface* surface) : Crée une texture à partir d'une surface
- int SDL_RenderCopy(SDL_Renderer* renderer, SDL_Texture* texture, const SDL_Rect* srcrect, const SDL_Rect* dstrect) : Copie une texture sur le rendu
- void SDL_RenderPresent(SDL_Renderer* renderer) : Met à jour l'affichage graphique avec le rendu
- int SDL_PollEvent(SDL_Event* event) : Met à jour la structure SDL_Event pour traiter l'évènement en attente

- void SDL_FreeSurface(SDL_Surface* surface) : Libère l'espace allouée à la surface
- int SDL_RenderClear(SDL_Renderer* renderer) : Nettoie le rendu
- void SDL_DestroyWindow(SDL_Window* window) : Détruit la fenêtre graphique
- void SDL_DestroyRenderer(SDL_Renderer* renderer) : Détruit le rendu
- void SDL_Quit() : Nettoie tous les éléments initialisés avec SDL
- const char* SDL_GetError(void) : Renvoie un message décrivant la dernière erreur s'étant produite
- Uint32 SDL_GetTicks(void) : Retourne le nombre de millisecondes écoulées depuis l'initialisation SDL
- void SDL_Delay(Uint32 ms) : Attendre pendant le temps spécifié en paramètre
- void Mix_CloseAudio() : Ferme et nettoie le mixer

Les Structures

- TTF_Font : Structure contenant les données de la police de caractère
- Mix_Music : Structure permettant de gérer les données liées au contenu audio

Les Fonctions

- void TTF_Init() : Initialise la police de caractère
- TTF_Font *TTF_OpenFont(const char *file, int ptsize) : Charge la police à la taille ptsize
- SDL_Surface *TTF_RenderText_Blended(TTF_Font *font, const char *text, SDL_Color fg) : Crée la surface du texte à partir de la police, du texte et de la couleur
- void TTF_CloseFont(TTF_Font *font) : Libère l'espace allouée à la police de caractère
- void TTF_Quit() : Ferme l'API lié à la librairie
- SDL_Surface *IMG_Load(const char *file) : Crée une surface à partir du fichier entré en paramètre
- int Mix_OpenAudio(int frequency, Uint16 format, int channels, int chunksize) : Initialise le mixer
- Mix_Music *Mix_LoadMUS(const char *file) : Charge la musique contenue dans le fichier placé en paramètre
- int Mix_PlayMusic(Mix_Music *music, int loops) : Joue la musique

TABLE DES MATIÈRES