

Références SDL2 pour eForth Windows

version 1.0 - lundi 28 octobre 2024



Auteur

- Marc PETREMANN

Contents

Auteur.....	1
SDL2.....	3
CreateRenderer window index flag -- render.....	3
CreateWindow zstr x y w h fl -- win.....	3
DestroyRenderer render -- fl.....	4
DestroyWindow win -- fl.....	4
GetError -- n.....	4
GetWindowFlags window -- win-flag.....	4
GetWindowSize windows *w *h -- fl.....	4
GetWindowSizeInPixels windows *w *h -- fl.....	4
Init n -- n.....	5
Quit --.....	5
RenderClear render -- 0 err.....	6
RenderPresent render --.....	6
SDL2.dll -- <name>.....	6
SDL_INIT_VIDEO -- n.....	6
SetRenderDrawColor renderer r g b a -- fl.....	6
SetWindowSize window w h --.....	6

SDL2

CreateRenderer **window index flag -- render**

Créer un contexte de rendu 2D pour une fenêtre.

Paramètres :

- **window** la fenêtre où le rendu est affiché
- **index** l'index du pilote de rendu à initialiser, ou -1 pour initialiser le premier prenant en charge les indicateurs demandés
- **flags** 0, ou un ou plusieurs SDL_RendererFlags combinés par OU.

```
variable WIN0

z" My first window with SDL2"
    X0_SCREEN_POSITION Y0_SCREEN_POSITION SCREEN_WIDTH SCREEN_HEIGHT
    SDL_WINDOW_SHOWN    SDL.CreateWindow WIN0 !

variable REN0

WIN0 @ -1 0 CreateRenderer REN0 !
```

CreateWindow **zstr x y w h fl -- win**

Créez une fenêtre avec la position, les dimensions et les indicateurs spécifiés.

Paramètres :

- **title** le titre de la fenêtre, en codage UTF-8
- **x** la position x de la fenêtre, SDL_WINDOWPOS_CENTERED ou SDL_WINDOWPOS_UNDEFINED
- **y** la position y de la fenêtre, SDL_WINDOWPOS_CENTERED ou SDL_WINDOWPOS_UNDEFINED
- **w** la largeur de la fenêtre, en coordonnées d'écran
- **h** la hauteur de la fenêtre, en coordonnées d'écran
- **flags** 0, ou un ou plusieurs SDL_WindowFlags combinés par OU

Renvoie **win** qui a été créé ou 0 en cas d'échec.

```
\ define size and position for SDL window
800 constant SCREEN_WIDTH
400 constant SCREEN_HEIGHT
```

```

200 constant X0_SCREEN_POSITION
50 constant Y0_SCREEN_POSITION

z" My first window with SDL2"
    X0_SCREEN_POSITION Y0_SCREEN_POSITION
    SCREEN_WIDTH SCREEN_HEIGHT
    SDL_WINDOW_SHOWN CreateWindow
    value WIN0

```

DestroyRenderer **render -- fl**

Détruit le contexte de rendu d'une fenêtre et libère les textures associées.

```

\ free ressources, end renderer and window
: freeRessources ( -- )
    REN0 DestroyRenderer drop
    WIN0 DestroyWindow drop
    Quit
;

```

DestroyWindow **win -- fl**

Détruit une fenêtre

```

\ WIN0 must be declared by value and set by CreateWindow
WIN0 DestroyWindow

```

GetError **-- n**

Récupère un message sur la dernière erreur survenue sur le fil actuel.

GetWindowFlags **window -- win-flag**

Récupère les flags de fenêtre.

GetWindowSize **windows *w *h -- fl**

Obtient la taille de la zone client d'une fenêtre.

GetWindowSizeInPixels **windows *w *h -- fl**

Récupère la taille d'une fenêtre en pixels.

Paramètres :

- **window** la fenêtre à partir de laquelle la taille du dessin doit être interrogée
- **w** un pointeur vers une variable pour stocker la largeur en pixels
- **h** un pointeur vers une variable pour stocker la hauteur en pixels

```

variable WIN.width
variable WIN.height
: draw ( -- )
    RENO 255 255 255 255 SetRenderDrawColor drop
    RENO RenderClear drop
    RENO RenderPresent drop
    WIN0 WIN.width WIN.height GetWindowSizeInPixels drop
;
draw
WIN.width @ . \ display: 800
WIN.height @ . \ display: 400

```

Init n -- n

Initialise la librairie SDL.

n doit être une valeur:

SDL_INIT_TIMER \ timer subsystem

SDL_INIT_AUDIO \ audio subsystem

SDL_INIT_VIDEO \ video subsystem; automatically initializes the events subsystem

SDL_INIT_JOYSTICK \ joystick subsystem; automatically initializes the events subsystem

SDL_INIT_HAPTIC \ haptic (force feedback) subsystem

SDL_INIT_GAMECONTROLLER \ controller subsystem; automatically initializes the joystick subsystem

SDL_INIT_EVENTS \ events subsystem

SDL_INIT_SENSOR

Renvoie 0 en cas de succès ou un code d'erreur négatif en cas d'échec. Appeler **GetError** pour plus d'informations.

```

\ Initialize SDL with error management
: SDL.init ( n -- )
    Init
    if
        ." SDL could not intialize! SDL_Error: " getError .
    then
;
SDL_INIT_VIDEO SDL.init

```

Quit --

Nettoie tous les sous-systèmes initialisés.

RenderClear **render -- 0 | err**

Efface la cible de rendu actuelle avec la couleur de dessin.

```
\ RENO is a value previously initialized with CreateRenderer
RENO RenderClear drop
```

RenderPresent **render --**

Mettre à jour l'écran avec tout rendu effectué depuis l'appel précédent.

SDL2.dll **-- <name>**

point d'entrée vers la librairie SDL2.dll

```
\ Destroy a window.
z" SDL_DestroyWindow"          1 SDL2.dll DestroyWindow ( window -- )
```

SDL_INIT_VIDEO **-- n**

Constante. Indique à la SDL que vous voulez initialiser le sous-système vidéo.

```
SDL_INIT_VIDEO SDL.Init
```

SetRenderDrawColor **renderer r g b a -- fl**

Définir la couleur utilisée pour les opérations de dessin (Rect, Ligne et Effacer)

Paramètres

- **renderer** le contexte de rendu
- **r** la valeur rouge utilisée pour dessiner sur la cible de rendu
- **g** la valeur verte utilisée pour dessiner sur la cible de rendu
- **b** la valeur bleue utilisée pour dessiner sur la cible de rendu
- **a** la valeur alpha utilisée pour dessiner sur la cible de rendu ; généralement **SDL_ALPHA_OPAQUE** (255). Utilisez **SetRenderDrawBlendMode** pour spécifier comment le canal alpha est utilisé

SetWindowSize **window w h --**

Définit la taille de la zone client d'une fenêtre.

Paramètres :

- **window** la fenêtre à modifier
- **w** la largeur de la fenêtre en pixels, dans les coordonnées de l'écran, doit être > 0

- **h** la hauteur de la fenêtre en pixels, dans les coordonnées de l'écran, doit être > 0

La taille de la fenêtre en coordonnées de l'écran peut différer de la taille en pixels. Utilisez **GetRendererOutputSize** pour obtenir la taille réelle de la zone client en pixels.

```
WIN0 400 200 SetWindowSize drop
```