

SDL2



# PROGRAMMATION D'UNE INTERFACE GRAPHIQUE



# *Simple DirectMedia Layer*

- Bibliothèque de développement multiplateforme
  - UNIX, LINUX, WINDOWS, MAC, Mobiles
- Ecrite en C, utilisable avec plein d'autres langages
- Née en 1998 et en version 2 depuis 2013
- Bas niveau aux périphériques et à la couche graphique
  - OpenGL et Direct3D
  - Greffons standards et moins standards
    - SDL2\_TTF, SDL2\_mixer, SDL2\_image, SDL2\_net, SDL2\_gfx
- Orientée événements
- Multifenêtres



# Bibliothèque multiplateforme

Votre application toute belle et toute innovante !

SDL

Module 1

Module 2

Xlib



Direct X



Autre système

Matériel



caché

# Bibliothèque bas niveau

- Formes basiques
  - Afficher points, lignes, rectangles et des textures en standard !
  - Ecrire à l'écran avec le greffon standard SDL2\_TTF
  - Lire des formats images autre que BMP avec SDL2\_Image
  - Afficher de ellipses dans le greffon non standard SDL2\_GFX
- Pas de composants hormis les fenêtres ou les boites de dialogues
- D'autres greffons standard : son, réseau



# Squelette d'une application SDL

- Initialisations graphiques
  - Variables globales ?
  - Structures
- Boucle principale = gestion des événements asynchrones
  - Attente d'un événement (bloquant ou non)
  - Gestion de l'événement
- Rendu des ressources graphiques



# File d'attente des événements

- Permet une gestion asynchrone des événements
- Lecture bloquante ou non d'un événement
- Eventuellement avec une limite
- Insertion d'événement système / système graphique / utilisateur
  
- Filtrage possible avant insertion
- Désactivation d'événements



# Types d'événements (1)

- Structure `SDL_Event` avec un champ *type*
- Système (notamment sur les mobiles)
- Clavier
- Souris
- Contrôleur / joystick / toucher / mouvement
- Moteur de rendu
- Audio
- Utilisateur
- Fenêtre →



# Types d'événements (2) - Fenêtre

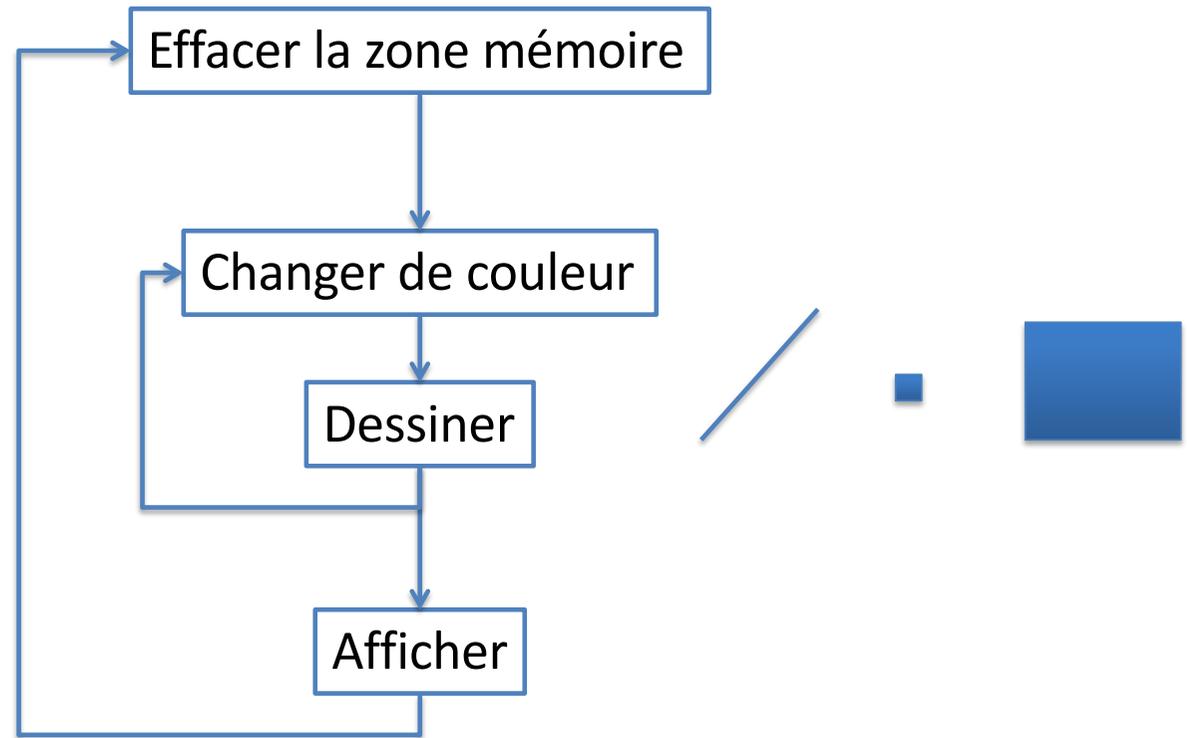
- Affichage / exposition

Les applications sont responsables de leur affichage !!!

- Focus clavier
- Entrée ou sortie de la souris
- Fermeture
- Réduction / restauration / maximisation
- Redimensionnement

# Contexte de moteur de rendu (*render*)

- Dessin 2D dans une zone mémoire-tampon (*buffer*)
- Affichage différé
- Exploitation des capacités des cartes graphiques



# Texture

- Zone dans laquelle on peut dessiner
  - Chargement d'images
  - Police de caractères (police TTF)
  - Import de surface (SDL 1.2)
  - Potentiellement gérée/accélérée par le matériel
- Cible temporaire d'un *render*
- Copie d'une texture vers un *render*

# Codes fournis

- Initialisation de la bibliothèque SDL2
- Exemple d'affichage d'un rectangle, d'un texte et d'un image
- Gestion de quelques événements

# SDL\_Event

```
typedef union SDL_Event
{
    Uint32 type;
    SDL_CommonEvent common;
    SDL_WindowEvent window;
    SDL_KeyboardEvent key;
    SDL_TextInputEvent text;
    SDL_MouseMotionEvent motion;
    SDL_MouseButtonEvent button;
    SDL_MouseWheelEvent wheel;
    SDL_JoyAxisEvent jaxis;
    SDL_QuitEvent quit;
    SDL_UserEvent user;
    SDL_TouchFingerEvent tfinger;
    // ...
} SDL_Event;
```

```
SDL_WaitEvent()
SDL_WaitEventTimeout()
SDL_PollEvent()
```



# Opérations binaires en C

OU  
|

0	0	0
0	1	1
1	0	1
1	1	1

Mettre à 1 les bits  
d'un masque en fonction de certains critères

ET  
&

0	0	0
0	1	0
1	0	0
1	1	1

OU EXCLUSIF  
^

0	0	0
0	1	1
1	0	1
1	1	0

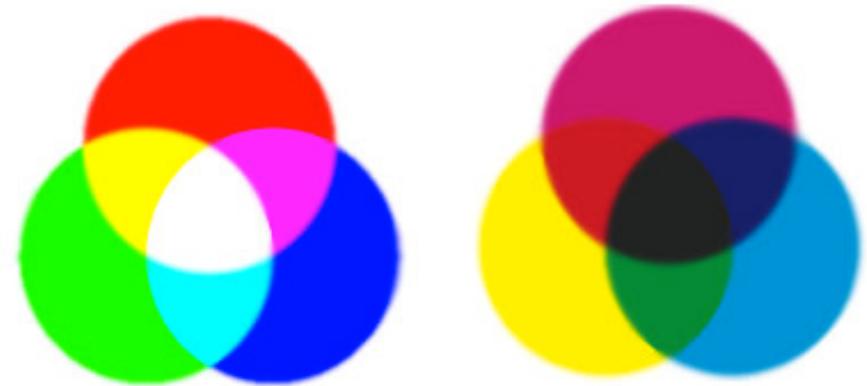
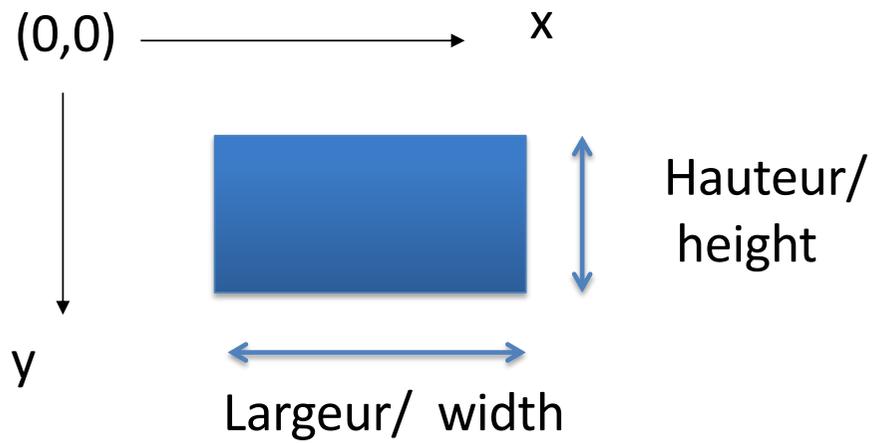
Afficher deux fois un pixel l'efface

# Exemples

- $5 \&\& 7$
- $5 \|\| 7$
- $!5$
- $5 \& 7$
- $5 | 7$
- $5 \ll 2$
- $5 \gg 2$
- $\sim 5$

$2^n$	64	32	16	8	4	2	1
5	0	0	0	0	1	0	1
7	0	0	0	0	1	1	1

# Dessiner



Couleurs : (ROUGE, VERT, BLEU, ALPHA)