Cours de C n°14 (intermédiaire)

Trimestre 2

Développement avec la bibliothèque SDL2 partie 1

Qu'est-ce que SDL ? À quoi sert SDL ?

- Simple DirectMedia Layer est une bibliothèque multimédia
- Elle permet d’accéder à l'audio, le clavier, la souris, le joystick et la carte graphique.
- Elle est utilisée dans des lecteurs vidéos, émulateurs, jeux populaires, etc...
- Elle est cross-platform (PC, mobiles et consoles)

Plateformes supportées

- Windows, Mac OS, Linux
- Android, iOS
- Dreamcast, Playstation 2, GP2x

Différences entre SDL1 et SDL2 ?

- accélération matérielle (gpu)
- multiples fenêtres
- pas de changements d'API pour l'audio (retro-compatibilité)
- Licence zlib
- ...

Initialiser SDL

Initialiser SDL :
int SDL_Init(int flags)

- SDL_Init(...) permet d'initialiser SDL
- retourne 0 si tout est ok
- retourne une valeur négative si une erreur

Initialiser seulement la vidéo

```c
// seulemement l'affichage : SDL_Init(SDL_INIT_VIDEO)

#include "SDL.h"
#include <stdio.h>

int main
{
        // Initialiser SDL (initialiser seulemement la video car on utilise seulement l'affichage)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // inserer le code SDL ici

        return 0;
}
```

Initialiser la vidéo + l'audio

```c
// affichage + audio : SDL_Init(SDL_INIT_VIDEO|SDL_INIT_AUDIO)

#include "SDL.h"
#include <stdio.h>

int main
{
        // Initialiser SDL (initialiser la vidéo/affichage + l'audio)
        if (SDL_Init(SDL_INIT_VIDEO|SDL_INIT_AUDIO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // inserer le code SDL ici

        return 0;
}
```

Initialiser tout

```c
// tout initialiser : SDL_Init(SDL_INIT_EVERYTHING)

#include "SDL.h"
#include <stdio.h>

int main
{
        // Initialiser SDL (initialiser tout)
        if (SDL_Init(SDL_INIT_EVERYTHING) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // inserer le code SDL ici

        return 0;
}
```

La fenêtre (window)

Créer une fenêtre

- SDL_Window* SDL_CreateWindow(const char* titre,
  int x,
  int y,
  int largeur,
  int hauteur,
  Uint32 flags)

- titre : titre de la fenêtre
- x, y : position de la fenêtre
- largeur, hauteur : taille de la fenêtre
- flags : les options qu'on veut utiliser

Détruire une fenêtre

- void SDL_DestroyWindow(SDL_Window* window)

- window → la fenêtre qu'on veut détruire

Afficher une fenêtre pendant 5 secondes (exemple)

```c
// creer une fenetre : SDL_CreateWindow(...)

#include "SDL.h"
#include <stdio.h>

int main(int argc, char** argv)
{
        SDL_Window* window = NULL;

        // Initialiser SDL (on utilise seulement la vidéo)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // Creer la fenetre
        window = SDL_CreateWindow("01 - Ma premiere fenetre SDL2",
                SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
                640, 480,
                SDL_WINDOW_SHOWN);

        if(window == NULL)
        {
                printf("Erreur pour creer la fenetre (%s)\n", SDL_GetError());
                return -1;
        }

        // attendre 5 secondes
        SDL_Delay(5000);

        // fin du programme
        SDL_DestroyWindow(window);
        SDL_Quit();
        return 0;
}
```

Afficher une fenêtre pendant 5 secondes

Attendre une durée fixe

Attendre N millisecondes

void SDL_Delay(Uint32 ms)

- ms : durée en millisecondes

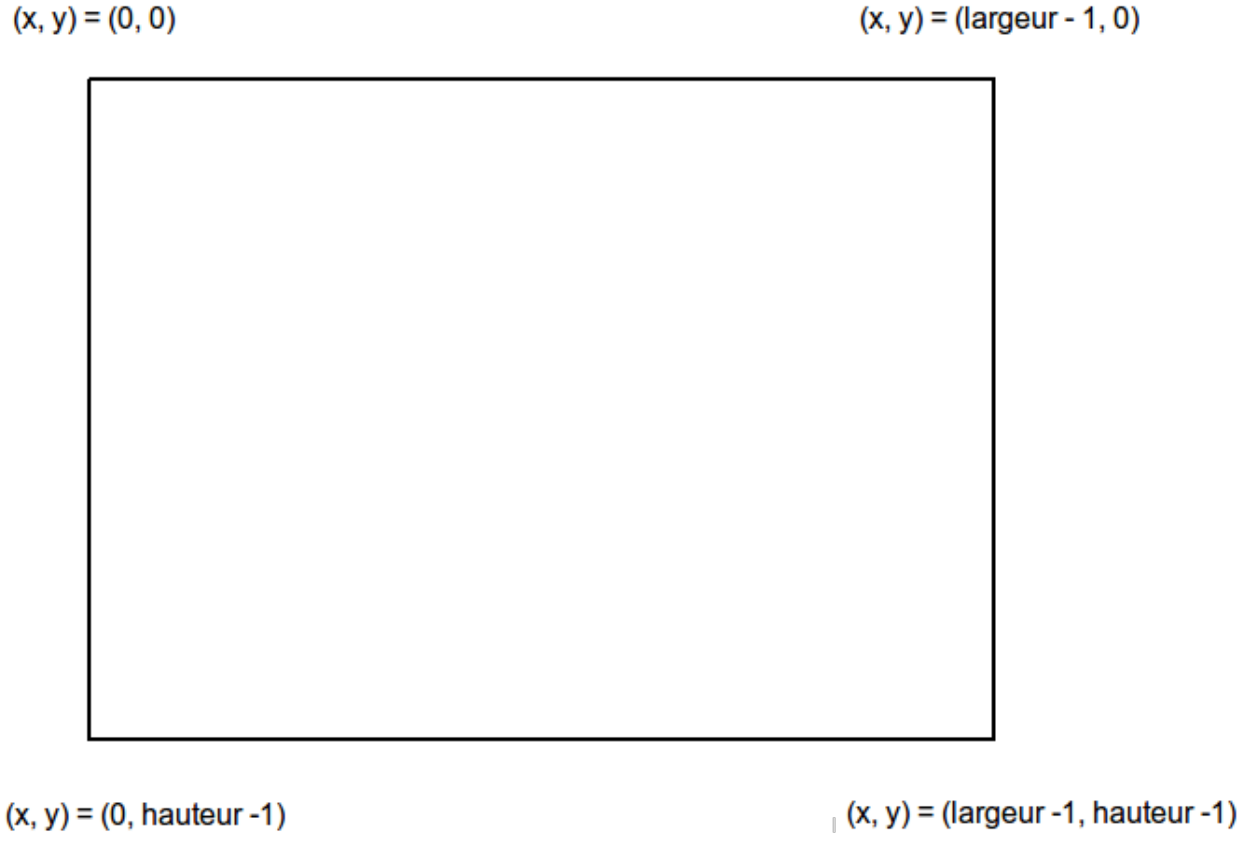
Attendre N millisecondes (exemples)

```
// attendre 1 seconde
SDL_Delay(1000);

// attendre 5 secondes
SDL_Delay(5000);

// attendre 1/4 seconde
SDL_Delay(250);
```

Dessiner avec SDL

Position écran

Le point (0,0) est toujours en haut à gauche pour dessiner



Quelques fonctions graphiques - 1/2

- créer le renderer : SDL_Renderer* SDL_CreateRenderer(SDL_Window* window, int index, Uint32 flags)
- changer la couleur de dessinage ou remplissage : SDL_SetRenderDrawColor(SDL_Renderer* renderer, int r, int g, int b, int a)
- effacer tout : SDL_RenderClear(SDL_Renderer* renderer)
- mettre à jour la fenêtre : SDL_RenderPresent(SDL_Renderer* renderer)

Quelques fonctions graphiques - 2/2

- dessiner un point : SDL_RenderDrawPoint(SDL_Renderer* renderer, int x, int y)
- dessiner N points : SDL_RenderDrawPoints(SDL_Renderer* renderer, const SDL_Point* points, int N)
- dessiner une ligne : SDL_RenderDrawLineSDL_Renderer* renderer, int x1, int y1, int x2,int y2)
- dessiner N lignes : SDL_RenderDrawLinesSDL_RenderDrawLines(SDL_Renderer* renderer, const SDL_Point* points, int N)
- dessiner un rectangle vide : SDL_RenderDrawRect(SDL_Renderer* renderer, const SDL_Rect* rect)
- dessiner N rectangles vides : SDL_RenderDrawRects(SDL_Renderer* renderer, const SDL_Rect* rects, int N)
- dessiner un rectangle rempli : SDL_FillRect(SDL_Renderer* renderer, const SDL_Rect* rect)
- dessiner N rectangles remplis : SDL_FillRects(SDL_Renderer* renderer, const SDL_Rect* rects, int N)

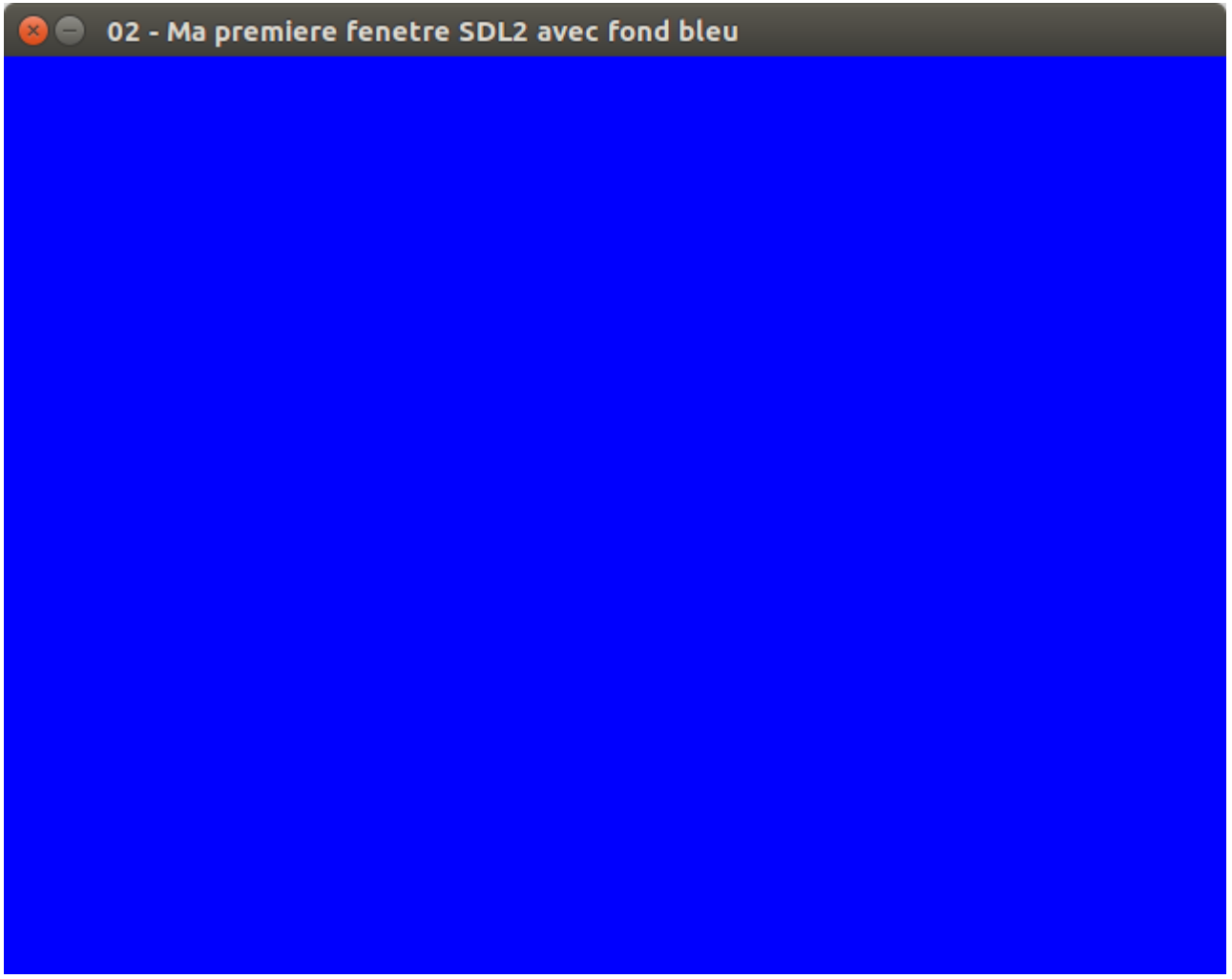
Afficher un arrière-plan bleu - 1/2

```c
#include "SDL.h"
#include <stdio.h>

int main(int argc, char** argv)
{
        SDL_Window* window = NULL;
        SDL_Renderer* renderer = NULL;

        // Initialiser SDL (on utilise seulement la vidéo)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // Creer la fenetre
        window = SDL_CreateWindow("02 - Ma premiere fenetre SDL2 avec fond blue",
                SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
                640, 480,
                SDL_WINDOW_SHOWN);
        if(window == NULL)
        {
                printf("Erreur pour creer la fenetre (%s)\n", SDL_GetError());
                return -1;
        }
```

Afficher un arrière-plan bleu - 2/2

```
        // creer le renderer
        renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
        if(renderer != NULL)
        {
                // effacer le contenu de la fenetre avec du blue
                SDL_SetRenderDrawColor(renderer, 0, 0, 255, 255);
                SDL_RenderClear(renderer);

                // mettre à jour l'ecran
                SDL_RenderPresent(renderer);
        }

        // attendre 5 secondes
        SDL_Delay(5000);

        // fin du programme
        SDL_DestroyWindow(window);
        SDL_Quit();
        return 0;
}
```

Afficher un arrière plan bleu



Charger une image et libérer une image

```
SDL_Surface* image = NULL;

// charger une image
image = SDL_LoadBMP("image.bmp");

// libérer une image
SDL_FreeSurface(image);
```

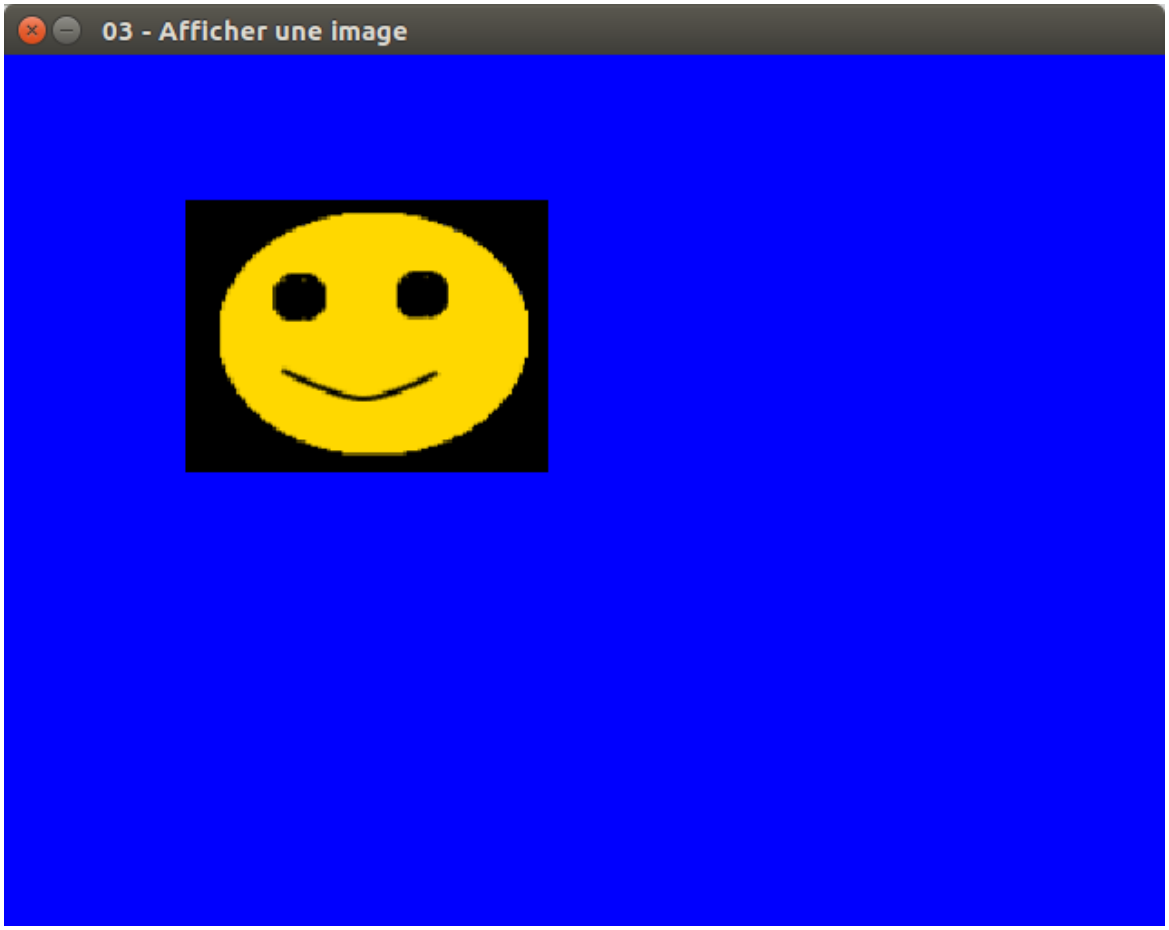
Charger + Convertir une image en texture pour l'afficher

```
SDL_Surface* image = NULL;
SDL_Texture* texture = NULL;

// charger une image
image = SDL_LoadBMP("image.bmp");

// convertir l'image en texture
texture = SDL_CreateTextureFromSurface(renderer, image);

// libérer une image
SDL_FreeSurface(image);

// effacer la texture
SDL_DestroyTexture(texture);
```

Afficher une image (texture)

```
// rectangle de l'image, il contient la position X, Y et la taille (largueur et hauteur)
SDL_Rect rectangle = { /*x*/100, /*y*/80, /*largeur*/200, /*hauteur*/150};

// afficher la texture/image à l'ecran
SDL_RenderCopy(renderer, texture, NULL, &rectangle);
```

Afficher une image (texture)



Afficher une image - 1/3 (exemple complet)

```c
#include "SDL.h"
#include <stdio.h>

int main(int argc, char** argv)
{
        int ret = 0;
        SDL_Window* window = NULL;
        SDL_Renderer* renderer = NULL;
        SDL_Surface* image = NULL;
        SDL_Texture* texture = NULL;

        // Initialiser SDL (on utilise seulement la vidéo)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // Creer la fenetre
        window = SDL_CreateWindow("03 - Afficher une image",
                SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
                640, 480,
                SDL_WINDOW_SHOWN);
        if(window == NULL)
        {
                printf("Erreur pour creer la fenetre (%s)\n", SDL_GetError());
                return -1;
        }
```

Afficher une image - 2/3 (exemple complet)

```
// creer le renderer
renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
if(renderer != NULL)
{
        // effacer le contenu de la fenetre avec du blue
        SDL_SetRenderDrawColor(renderer, 0, 0, 255, 255);
        SDL_RenderClear(renderer);

        // charger l'image
        image = SDL_LoadBMP("image.bmp");
        if(image != NULL)
        {
                // rectangle contient la position X, Y et la taille (largueur et hauteur)
                SDL_Rect rectangle = { /*x*/100, /*y*/80, /*largeur*/200, /*hauteur*/150};

                // convertir l'image en texture
                texture = SDL_CreateTextureFromSurface(renderer, image);

                // effacer l'image car on n'en a plus besoin car la texture contient l'image
                SDL_FreeSurface(image);

                // afficher la texture/image à l'ecran
                SDL_RenderCopy(renderer, texture, NULL, &rectangle);

                // effacer la texture car on en a plus besoin une fois affichée
                SDL_DestroyTexture(texture);
        }
        else
        {
                ret = -1;
                printf("Erreur de chargement de l'image (%s)\n", SDL_GetError());
        }


        // mettre à jour l'ecran
        SDL_RenderPresent(renderer);
}
```

Afficher une image - 3/3 (exemple complet)

```
        else
        {
                ret = -1;
                printf("Erreur pour creer le renderer (%s)\n", SDL_GetError());
        }

        // attendre 5 secondes
        SDL_Delay(5000);

        // fin du programme
        SDL_DestroyWindow(window);
        SDL_Quit();
        return ret;
}
```

Les événements (events)

Qu'est-ce qu'un événement (event) ?

- Un événement est une action utilisateur ou de l'OS
- Ils sont stockés dans une file/queue : liste qui conserve l'ordre d'arrivé (FIFO : First In First Out = 1er arrivé 1er servi)
- Quelques exemples d'évenement : Touche du clavier préssée, Click de la souris, redimensionnement de la fenêtre, ...

Quelques types d'événements en SDL :

- SDL_QUIT : L'utilisateur veut fermer l'application SDL
- SDL_KEYDOWN : Une touche du clavier est préssée
- SDL_KEYUP : Une touche du clavier est relachée
- SDL_MOUSEBUTTONDOWN : Un bouton de la souris est préssé
- SDL_MOUSEBUTTONUP : Un bouton de la souris est relaché

Lire tous les événements

```c
int quit = 0;
SDL_Event event;

// lire tous les evenements
while (SDL_PollEvent(&event))
{
        // Test le type de l'évenement : SDL_QUIT, SDL_KEYDOWN, SDL_KEYUP, SDL_MOUSEBUTTONDOWN, SDL_MOUSEBUTTONUP
        switch (event.type)
        {
                case SDL_QUIT:
                {
                        printf("reception de l evenement SDL_QUIT\n");
                        quit = 1;
                        break;
                }
                case SDL_KEYDOWN:
                {
                        printf("touche %d du clavier préssée\n", event.key.keysym.sym);
                        break;
                }
                case SDL_MOUSEBUTTONDOWN:
                {
                        printf("bouton %d de souris préssée\n", event.button.button);
                        break;
                }
        }
}
```

Gérer l'événement "Quitter"

```
// L'évenement SDL_QUIT est envoyé quand on clique sur le "X" de la fenetre

int quit = 0;
SDL_Event e;

// boucle principale
while(!quit)
{
        // lire tous les evenements
        while (SDL_PollEvent(&e))
        {
                // stopper la boucle principale si évenememement est SDL_QUIT
                if (e.type == SDL_QUIT)
                {
                        printf("reception de l evenement SDL_QUIT\n");
                        quit = 1;
                }
        }

        // Mettre ici votre code d'affichage SDL
        // ...
}
```

Gérer l'événement "Quitter" – 1/3 (exemple complet)

```c
#include "SDL.h"
#include <stdio.h>

int main(int argc, char** argv)
{
        SDL_Window* window = NULL;
        SDL_Renderer* renderer = NULL;

        // Initialiser SDL (on utilise seulement la vidéo)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // Creer la fenetre
        window = SDL_CreateWindow("04 - Gestion de l'evenement SDL_QUIT",
                SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
                640, 480,
                SDL_WINDOW_SHOWN);
        if(window == NULL)
        {
                printf("Erreur pour creer la fenetre (%s)\n", SDL_GetError());
                return -1;
        }
```

Gérer l'événement "Quitter" – 2/3 (exemple complet)

```c
// creer le renderer
renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
if(renderer != NULL)
{
        int quit = 0;
        SDL_Event e;

        // boucle principale
        while(!quit)
        {
                // lire tous les evenements
                while (SDL_PollEvent(&e))
                {
                        // stopper la boucle principale si évenememement est SDL_QUIT
                        if (e.type == SDL_QUIT)
                        {
                                printf("reception de l evenement SDL_QUIT\n");
                                quit = 1;
                        }
                }
                // effacer le contenu de la fenetre avec du blue
                SDL_SetRenderDrawColor(renderer, 0, 0, 255, 255);
                SDL_RenderClear(renderer);

                // mettre à jour l'ecran
                SDL_RenderPresent(renderer);

                // attendre 50 millisecondes
                SDL_Delay(50);
        }
}
```

Gérer l'événement "Quitter" – 3/3 (exemple complet)

```
        // fin du programme
        SDL_DestroyWindow(window);
        SDL_Quit();
        return 0;
}
```

Utiliser le clavier

Il faut gérer les événements de type SDL_KEYDOWN (touche préssée) ou SDL_KEYUP (touche relachée)

Utiliser le clavier

```
SDL_Event evenement;

while(!quit)
{
        // lire tous les evenements
        while (SDL_PollEvent(&evenement))
        {
                // Test le type de l'évenement : SDL_QUIT, SDL_KEYDOWN, SDL_KEYUP, SDL_MOUSEBUTTONDOWN, SDL_MOUSEBUTTONUP
                switch (evenement.type)
                {
                        case SDL_KEYDOWN:
                        {
                                printf("touche %d du clavier préssée\n", evenement.key.keysym.sym);
                                if (evenement.key.keysym.sym == SDLK_a)
                                {
                                        // gérer la touche A
                                }
                                else if (evenement.key.keysym.sym == SDLK_b)
                                {
                                        // gérer la touche B
                                }
                                else if (evenement.key.keysym.sym == SDLK_2)
                                {
                                        // gérer la touche 2
                                }
                                break;
                        }
                }
        }
}
```

Utiliser le clavier – 1/3 (exemple complet)

```c
#include "SDL.h"
#include <stdio.h>

int main(int argc, char** argv)
{
        SDL_Window* window = NULL;
        SDL_Renderer* renderer = NULL;

        // Initialiser SDL (on utilise seulement la vidéo)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // Creer la fenetre
        window = SDL_CreateWindow("05 - Gestion du clavier : appuyer sur + or - du numpad",
                SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
                640, 480,
                SDL_WINDOW_SHOWN);
        if(window == NULL)
        {
                printf("Erreur pour creer la fenetre (%s)\n", SDL_GetError());
                return -1;
        }

        // creer le renderer
        renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
        if(renderer != NULL)
```

Utiliser le clavier – 2/3 (exemple complet)

```
{
        int color = 100;
        int quit = 0;
        SDL_Event evenement;

        while(!quit)
        {
                // lire tous les evenements
                while (SDL_PollEvent(&evenement))
                {
                        // Test le type de l'évenement : SDL_QUIT, SDL_KEYDOWN, SDL_KEYUP, SDL_MOUSEBUTTONDOWN, SDL_MOUSEBUTTONUP
                        switch (evenement.type)
                        {
                                case SDL_QUIT:
                                {
                                        printf("reception de l evenement SDL_QUIT\n");
                                        quit = 1;
                                        break;
                                }
                                case SDL_KEYDOWN:
                                {
                                        printf("touche %d du clavier préssée\n", evenement.key.keysym.sym);
                                        if (evenement.key.keysym.sym == SDLK_KP_PLUS)
                                        {
                                                color += 10;
                                                if (color > 255) color = 255;
                                        }
                                        else if (evenement.key.keysym.sym == SDLK_KP_MINUS)
                                        {
                                                color -= 10;
                                                if (color < 0) color = 0;
                                        }
                                        break;
                                }
                        }
                }
```

Utiliser le clavier – 3/3 (exemple complet)

```
                // effacer le contenu de la fenetre avec du blue
                SDL_SetRenderDrawColor(renderer, color, color, color, 255);
                SDL_RenderClear(renderer);

                // mettre à jour l'ecran
                SDL_RenderPresent(renderer);

                // attendre 50 millisecondes
                SDL_Delay(50);
        }
}

// fin du programme
SDL_DestroyWindow(window);
SDL_Quit();
return 0;
```

Utiliser la souris

Il faut gérer les événements de type SDL_MOUSEBUTTONDOWN (bouton de la souris préssé) ou SDL_MOUSEBUTTONUP (bouton de la souris relaché)

Info supplémentaire : evenement.motion.x et evenement.motion.y donne la position de la souris

Utiliser la souris (exemple)

```
SDL_Event evenement;

// lire tous les evenements
while (SDL_PollEvent(&evenement))
{
        // Test le type de l'évenement : SDL_QUIT, SDL_KEYDOWN, SDL_KEYUP, SDL_MOUSEBUTTONDOWN, SDL_MOUSEBUTTONUP
        switch (evenement.type)
        {
                case SDL_MOUSEBUTTONDOWN:
                {
                        printf("bouton %d de souris préssée\n", evenement.button.button);
                        printf("position de la souris (%d, %d)\n", evenement.motion.x, evenement.motion.y);
                        break;
                }
        }
}
```

Utiliser la souris – 1/3 (exemple complet)

```c
#include "SDL.h"
#include <stdio.h>

int main(int argc, char** argv)
{
        SDL_Window* window = NULL;
        SDL_Renderer* renderer = NULL;

        // Initialiser SDL (on utilise seulement la vidéo)
        if (SDL_Init(SDL_INIT_VIDEO) != 0)
        {
                printf("Erreur pour initialiser SDL (%s)\n", SDL_GetError());
                return -1;
        }

        // Creer la fenetre
        window = SDL_CreateWindow("06 - Gestion de la souris : cliquez pour afficher les infos",
                SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
                640, 480,
                SDL_WINDOW_SHOWN);
        if(window == NULL)
        {
                printf("Erreur pour creer la fenetre (%s)\n", SDL_GetError());
                return -1;
        }

        // creer le renderer
        renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);
        if(renderer != NULL)
```

Utiliser la souris – 2/3 (exemple complet)

```
{
        int quit = 0;
        SDL_Event evenement;

        while(!quit)
        {
                // lire tous les evenements
                while (SDL_PollEvent(&evenement))
                {
                        // Test le type de l'évenement : SDL_QUIT, SDL_KEYDOWN, SDL_KEYUP, SDL_MOUSEBUTTONDOWN, SDL_MOUSEBUTTONUP
                        switch (evenement.type)
                        {
                                case SDL_QUIT:
                                {
                                        printf("reception de l evenement SDL_QUIT\n");
                                        quit = 1;
                                        break;
                                }
                                case SDL_MOUSEBUTTONDOWN:
                                {
                                        printf("bouton %d de souris préssée\n", evenement.button.button);
                                        printf("position de la souris (%d, %d)\n", evenement.motion.x, evenement.motion.y);
                                        break;
                                }
                        }
                }

                // effacer le contenu de la fenetre avec du vert
                SDL_SetRenderDrawColor(renderer, 0, 255, 0, 255);
                SDL_RenderClear(renderer);
```

Utiliser la souris – 3/3 (exemple complet)

```
                // effacer le contenu de la fenetre avec du vert
                SDL_SetRenderDrawColor(renderer, 0, 255, 0, 255);
                SDL_RenderClear(renderer);

                // mettre à jour l'ecran
                SDL_RenderPresent(renderer);

                // attendre 50 millisecondes
                SDL_Delay(50);
            }
        }

        // fin du programme
        SDL_DestroyWindow(window);
        SDL_Quit();
        return 0;
}
```

La boucle principale ou la gameloop

La boucle principale ou la gameloop en 4 étapes:

- (1) Event - boucle pour lire/gérer tous les évenements via SDL_PollEvent()
- (2) Update - Faire nos calculs
- (3) Draw - Afficher l' "écran" de la fenetre + mettre à jour à la fin via SDL_RenderPresent()"
- (4) Wait - Attendre une petite durée

Gérer un framerate constant (version simplifiée)

- Par exemple, pour synchroniser l'affichage de l'application SDL avec un framerate fixe de 20 FPS (20 images par secondes), il suffit d’attendre : 1 secondes / 20

  1 secondes / 20 = 1000 millisecondes / 20 = 50 ms

- Remarque : on considère que les autres étapes prennent 0 millisecondes (pas toujours vrais)

La gameloop avec un framerate constant de 20 FPS (version simplifiée)

```
while (!quit)
{
        // (1) - Gérer les évenements
        // (2) - Faire nos calculs
        // (3) - Afficher

        // (4) - Attendre
        #define FPS 20
        SDL_Delay(1000 / FPS);
}
```