

Programmation C TP 7

L'objectif de ce TP est de programmer un jeu de sudoku avec la librairie graphique C de l'université : la libMLV.

Vous pouvez facilement partir du code de l'ancien TP (TP6) histoire de ne pas partir de rien. Il n'est absolument pas nécessaire de posséder une fonction solver pour résoudre les grilles, c'est l'utilisateur qui devra résoudre la grille.

Dans un soucis de simplicité, on tentera de produire un jeu qui se joue entièrement à la souris, la libMLV possède toutefois des fonctions avancées de gestion des événements clavier/souris. Pour ce TP, toutes les interactions de l'utilisateur seront récupérables avec la fonction `MLV_wait_mouse` qui attend un clic de la souris et met à jour deux pointeurs vers des entiers qui donnent les coordonnées de la souris au moment du clic.

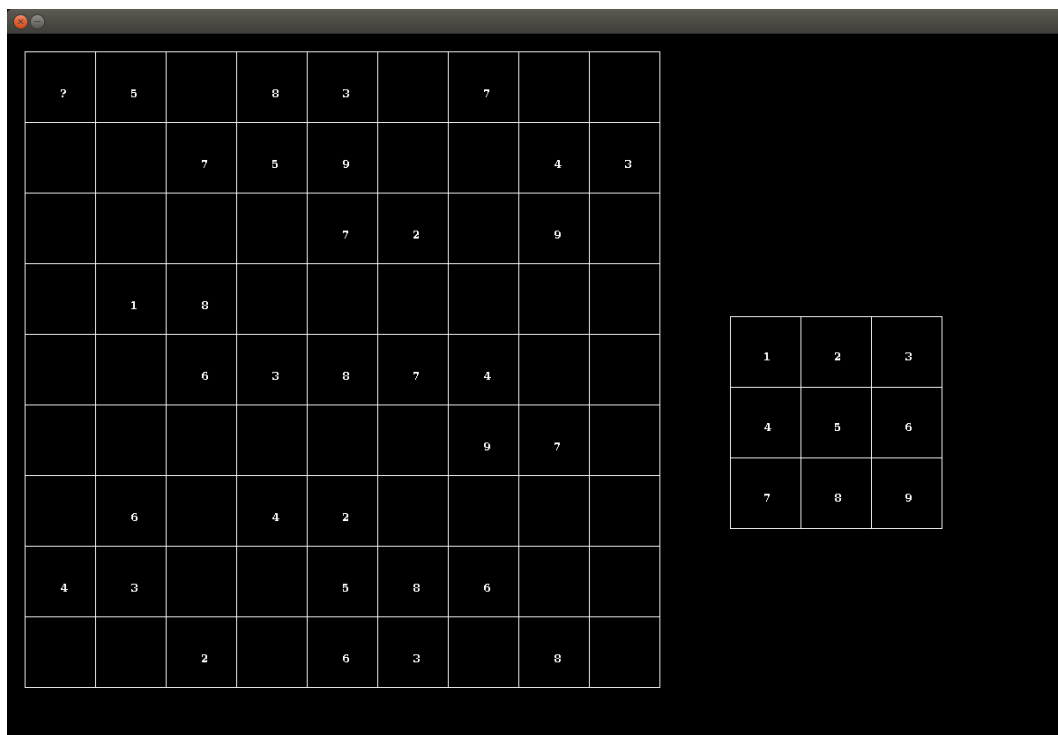
Pour vous donner une idée de rendu graphique (n'hésitez pas à vous exprimer et prendre de la liberté par rapport à ce modèle), voici un exemple avec 10 lignes verticales, 10 lignes horizontales et les numéros non nuls de la grille du fichier `data/grid1.txt` du TP6.

	5		8	3		7		
		7	5	9			4	3
				7	2		9	
	1	8						
		6	3	8	7	4		
						9	7	
	6		4	2				
4	3			5	8	6		
		2		6	3		8	

L'affichage est géré pixel par pixel dans la fenêtre. Tous les objets doivent être décrits avec des points dont il faut donner les coordonnées. Le coin supérieur gauche est le point de coordonnées (0,0). Le prototype qui illustre cet énoncé utilise une fenêtre de taille 1200 par 800. Il y a donc 1200 pixels pour la largeur et 800 en hauteur. Le point en bas à droite de la fenêtre a ainsi pour coordonnées (1199,799).

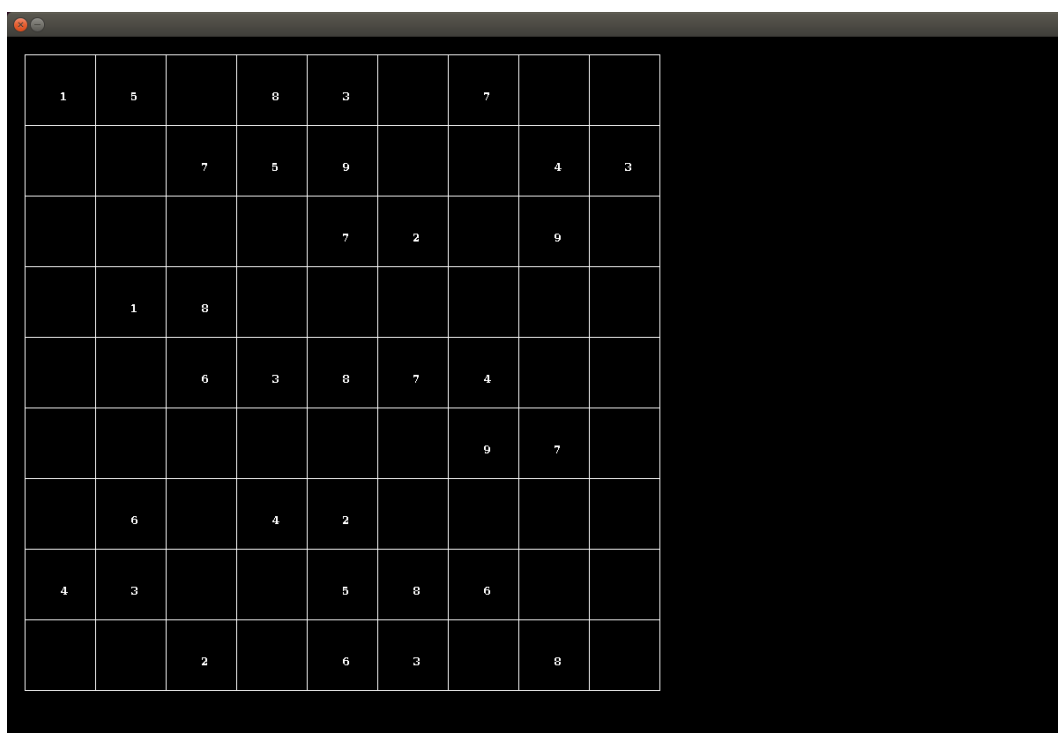
L'utilisateur doit pouvoir cliquer une case vide (celle comportant un 0 dans le Board en mémoire). Lorsque l'utilisateur clique sur une case remplissable, on lui affiche un cadran

pour qu'il puisse désigner avec la souris quelle valeur il souhaite insérer. Voici ce que cela peut donner en cliquant dans la case en haut à gauche du sudoku.



On a pris soin ici de placer un caractère ? dans la case à remplir pour rappeler à l'utilisateur quelle case il s'apprête à renseigner.

Si l'utilisateur choisit une valeur légale (c'est à dire une valeur qui n'enfreint pas les règles du sudoku), on remplit la case avec cette valeur et on affiche de nouveau la grille pour attendre un nouvel ordre de remplissage.



La partie se termine si l'utilisateur arrive à remplir complètement la grille.

Quelques rappels/conseils pour coder avec la libMLV :

- Faire un module graphique dédié à la partie graphique (avec un fichier de code et un fichier d'entêtes).
- Inclure les headers de la libMLV dans le fichier de code : `#include <MLV/MLV_all.h>`. Lorsque le fichier d'entête comporte des types de la libMLV, faire cette include dans le fichier d'entête du module graphique.
- Compiler votre module graphique et votre exécutable avec le flag supplémentaire `-lMLV`. Le flag doit donc apparaître au moins deux fois dans votre makefile.

Voici la liste des fonctions issues de la LibMLV utilisées par vos enseignants pour réaliser un prototype de votre TP :

- `MLV_create_window` (création d'une fenêtre)
- `MLV_free_window` (libération de la fenêtre)
- `MLV_actualise_window` (demande d'opérer les changements d'affichage à l'écran)
- `MLV_draw_text` (afficher un texte)
- `MLV_draw_filled_rectangle` (pour tout effacer, on met un rectangle noir dans la fenêtre)
- `MLV_rgba` (fabrication d'une couleur)
- `MLV_draw_line` (dessine une ligne en fonction des coordonnées de deux points et d'une couleur)
- `MLV_wait_mouse` (attends un clic de la souris dans la fenêtre et donne les coordonnées de ce clic)

Suggestions d'amélioration :

- Changer la font d'affichage des numéros (augmentez la taille surtout).
- Faire un mécanisme pour annuler des placements de nombres. L'utilisateur peut ainsi annuler des actions.
- Mettre en compteur pour calculer le temps de jeu.
- Faire un tableau des high scores avec les meilleurs temps.
- (très difficile) mettre en surbrillance (changement de couleur) la ligne et la colonne où se trouve la souris.
- Tout ce qui vous paraît sympathique à rajouter...