

RAPPORT TP1

I) Introduction

Ce TP 1 nous sert d'introduction à la librairie **NCURSES**, il va nous apprendre à nous familiariser à cette nouvelle librairie et à savoir comment l'utiliser pour nos futur **TP** et **PROJETS**.

II) Réponses aux questions

Exercice 1 :

1.

(a) Le terminal nous retourne cela :



Le « **print** » étant la première ligne tapée à l'intérieur de la fenêtre on peut donc en déduire que la **position d'origine** est en haut à gauche.

Quant à la commande « **move** », elle demande deux arguments (x, y) pour déplacer le « **curseur** » sur les positions au préalable écrites par le développeur.

(b) Les entités **LINES** et **COLS** donnent le nombre de lignes et nombre de colonnes qu'il y a sur un affichage de votre terminal.

(c) La commande « **move** », demande deux arguments (x, y) pour déplacer le « **curseur** » sur les positions au préalable écrites par le développeur.

(d) Comparaison et explication des fonctions :

- « **addch** » : Ajoute un caractère à la position courante et avance le curseur.
- « **mvaddch** » : Déplace la position courante et ajoute un caractère.
- « **printw** » : Ajoute une chaîne de caractère à la position courante et avance le curseur.
- « **mvprintw** » : Déplace la position courante et ajoute une chaîne de caractère.

2.

(a) La fonction « **attron** » sert à modifier le graphisme du texte, en le mettant en « gras », « souligné » ...

(b) */* Retire l'inversion de couleur du texte */*

« **attroff** » sert à retirer / désactiver un paramètre de texte activé avec « **attron** ».

3.

(a) */* Ecrire la chaîne de caractère « Abc123 ** * » en rouge à fond cyan à une position 2, 3. */*

(b) Il suffit de changer la ligne 8 en cette ligne :

« **init_pair(2, COLOR_GREEN, COLOR_BLUE);** »

Exercice 2 :

1.

(a) La fonction « **clear** » permet de vider la fenêtre de terminal pour enfin faire réapparaître le 'o' à un autre endroit.

(b) En mettant le « **clear** » entre le « **mvaddch** » et le « **refresh** » on retrouve un terminal vide.

On peut expliquer cela par le fait que comme on « **clear** » avant le « **refresh** » rien ne sera visible.

(c) Ne pas mettre le « **refresh** » avec le « **mvaddch** » car le même problème se produira.

La fonction « **usleep** » permet de stopper le programme pendant (x) **microsecondes**.

Dans le cas de notre programme, elle sert à donner cet aspect de « **dessin animé** ».

2.

(a) Le « **clear** » n'est pas la bonne option car elle ne rend pas fluide l'affichage du programme.

(d) En déplaçant le « **refresh** » plus rien n'apparaît.

Exercice 3 :

1.

(a) Décrire le rôle des fonctions « **getstr** » et « **mvscanw** » ;

- « **getstr** » : récupère une chaîne de caractère écrite par l'utilisateur.

- « **mvscanw** » : récupère ce qu'écrit l'utilisateur (un **int** ou **str** ou ...)

2.

(a) Commenter la ligne 9 et en déduire le rôle de la fonction « **noecho** » ;

Met les « **echo** » sur off, c'est à dire que les touches saisies par l'utilisateur n'apparaîtront pas sur le terminal.

3.

(a) Commenter la ligne 9 et en déduire le rôle de l'appel à la fonction « **nodelay** » avec les arguments « **stdscr** » (écran standard) et « **TRUE** ». (Indice : ceci modifie le mode de comportement de la fonction « **getch** ».) ;

Cette fonction permet d'indiquer qu'il ne faut pas bloquer l'appel de la fonction « **getch** » lorsqu'il n'y a pas de caractère de disponible.

Exercice 4 :

1.

(a) Commenter la ligne 31 et en déduire le rôle de l'appel à la fonction « **mousemask** » avec les arguments mentionnés.

Configure les événements de souris à surveiller dans ce cas-là tous les événements de la souris !

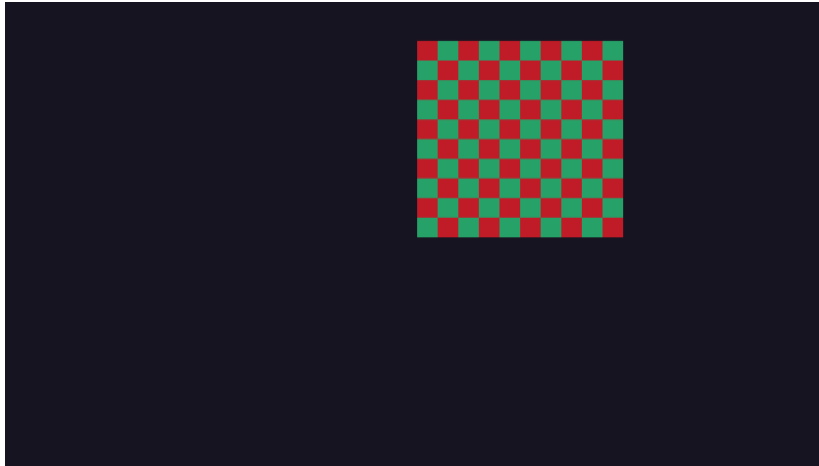
(b) Modifier le programme afin qu'il affiche, lorsque la souris attrape le chat, le message "**Attrape !**" au centre de la fenêtre. Après exactement **500 ms**, le message disparaît et un nouveau chat apparaît ensuite.

Il nous a suffi de clear la fenêtre et d'afficher un message au milieu grâce à 'mvprintw'
Et d'ensuite « **usleep 500000** » pour « **reclear** » la fenêtre.

III) Notions dans la pratique

Exercice 5 : (Damier)

Utilisation des fonctions « **init_pair / start_color** » de l'exercice 1.3 pour pouvoir reproduire un damier au milieu.



Exercice 6 : (Triangle)

Simple exercice qui avait été fait en python l'année passée. Il suffit seulement de faire une « **double boucle for** », pour pouvoir faire un triangle.

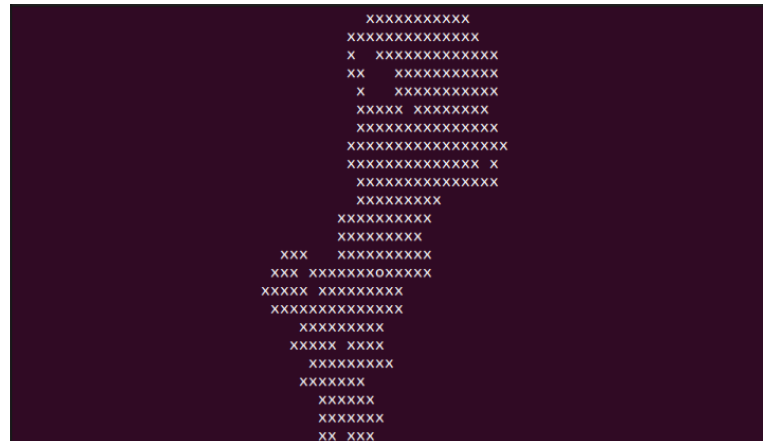


Exercice 7 : (Marche aléatoire)

Utilisation du module « **time et stdlib** » pour pouvoir utiliser la fonction « **rand et srand** ».
Partons du milieu du terrain, soit « **COLS/2** » et « **LINES/2** ».

Dans une « **boucle while** », il suffit par saisie contrôlée de créer des positions orthogonales du points au milieu pour déplacer le « **o** » et mettre un « **x** » à son ancienne place.

Pour gérer la vitesse et le fait de mettre en pause, on utilise « **touche = getch() ;** » comme dans les exercices précédents.

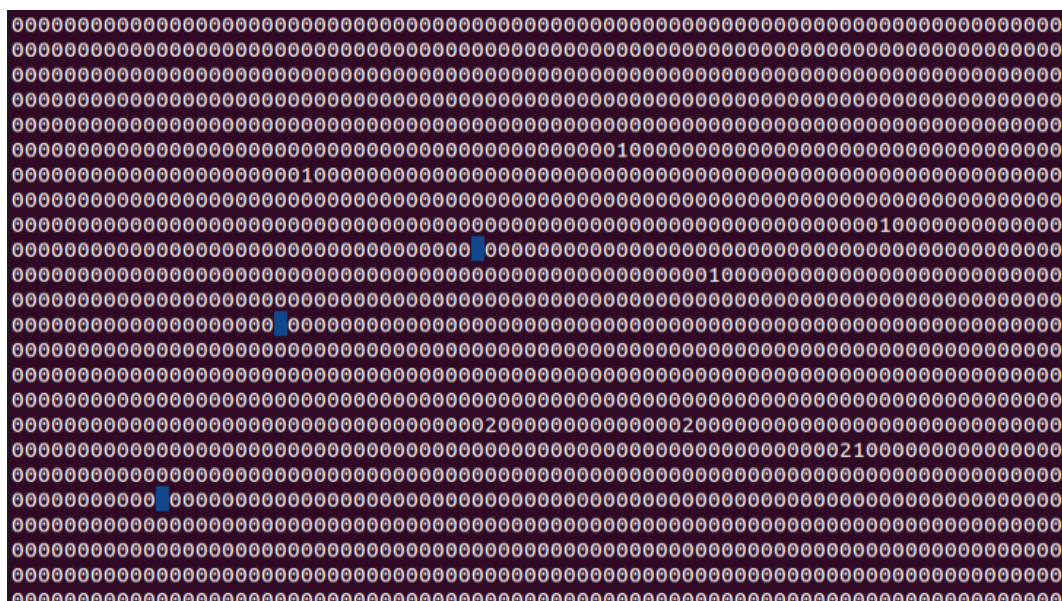


Exercice 8 : (Clics)

Pour le **dernier** exercice, nous avons favorisé la création d'un tableau à deux dimensions **COLS** et **LINES** pour suivre l'avancement des clics et des valeurs sur chacune des positions.

Dans une boucle **WHILE**, on demande à l'utilisateur de cliquer, on regarde dans le tableau à quelle valeur correspond les coordonnées si c'est « **0** » on met à « **1** », à « **2** » si c'est « **1** » et en **bleu** (grâce à un exercice au-dessus) si c'est « **2** ».

Sans oublier une fonction qui remplit de « **0** » tout le terminal.



IV) Conclusion

1) Difficultés rencontrées

Dans ce **TP 1** nous n'avons pas rencontré de difficultés particulières.

2) Ce que le sujet nous a apporté

Ce **TP 1** nous a appris à nous servir de la librairie **NCURSES** qui nous servira pour nos futurs **TP** et **PROJETS**

ANNEXE 1

Lors de la première exécution il faut entrer la commande :

```
chmod u+x nccomp
```

Cette commande permet que le **nccom** soit reconnu.

Pour compiler le **projet** il faut exécuter la commande suivante :

```
./nccomp nom_du_fichier
```

ANNEXE 2

Pour utiliser le **projet** il suffit d'exécuter la commande suivante :

```
./nom_du_fichier
```