

ncurses

Gestion de la souris avec ncurses

Cet article présente comment il est possible de récupérer les évènements de la souris (type de clic, position) avec ncurses.

1 Activation de la souris

Le fichier `ncurses.c` contient plusieurs fonctions nécessaires pour `ncurses` : l'activation et l'arrêt du mode `ncurses`, l'activation et la configuration des couleurs, l'activation de la souris et la récupération de la position de la souris suite à un évènement (un clic par exemple).

La fonction `ncurses_souris` active la souris et vérifie qu'une souris est détectée. Elle doit être exécutée une fois le mode `ncurses` démarré (avec la fonction `ncurses_initialiser`).

```
void ncurses_souris() {
    if(!mousemask(ALL_MOUSE_EVENTS, NULL)) {
        ncurses_stopper();
        fprintf(stderr, "Erreur_lors_de_l'initialisation_de_la_souris.\n");
        exit(EXIT_FAILURE);
    }

    if(has_mouse() != TRUE) {
        ncurses_stopper();
        fprintf(stderr, "Aucune_souris_n'est_détectée.\n");
        exit(EXIT_FAILURE);
    }
}
```

Lorsque la fonction `getch` détecte un évènement souris, nous avons besoin de récupérer cet évènement (fonction `getmouse`) et éventuellement la position de la souris à l'écran. La fonction `souris_getpos` permet de réaliser ces actions :

```
int souris_getpos(int *x, int *y, int *bouton) {
    MEVENT event;
    int resultat = getmouse(&event);

    if(resultat == OK) {
        *x = event.x;
        *y = event.y;
        *bouton = event.bstate;
    }
    return resultat;
}
```

2 Le programme de test

Dans ce programme, nous commençons par appeler les deux fonctions pour activer le mode ncurses et la souris. Nous activons également le défilement dans la fenêtre principale (stdscr) avec la fonction `scrollok`.

```
ncurses_initialiser();
ncurses_souris();
scrollok(stdscr, TRUE);
```

Dans la boucle principale, nous attendons que l'utilisateur clique avec la souris. La fonction `getch` retourne la constante `KEY_MOUSE`. Nous utilisons ensuite la fonction `souris_getpos` pour récupérer la position et les boutons pressés.

```
while((ch = getch()) != KEY_F(2)) {
    switch(ch) {
        case KEY_MOUSE:
            if(souris_getpos(&sourisX, &sourisY, &bouton) == OK) {
                if(bouton & BUTTON1_CLICKED)
                    printf("Clic_bouton_1_a_la_position_(%d,%d)\n", sourisX,
                        , sourisY);
                if(bouton & BUTTON2_CLICKED)
                    printf("Clic_bouton_2_a_la_position_(%d,%d)\n", sourisX,
                        sourisY);
                if(bouton & BUTTON1_DOUBLE_CLICKED)
                    printf("Double-clic_bouton_1_a_la_position_(%d,%d)\n",
                        sourisX, sourisY);
                if(bouton & BUTTON2_DOUBLE_CLICKED)
                    printf("Double-clic_bouton_2_a_la_position_(%d,%d)\n",
                        sourisX, sourisY);
                refresh();
            }
    }
}
```

3 Compilation et exécution

Le `makefile` fourni permet de compiler le programme précédent. Tapez la commande suivante :

```
make
```

Pour exécuter le programme, tapez la commande suivante :

```
./exemple
```