

```

#include <ncurses.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include "pconstante.h"
#include "prototype.h"

static void finish(int sig);

typedef struct
{
    char nom[7];          // nom joueur
    int l;               //position en ligne
    int c;               // position colonne
    int couleur ;       // couleur associee
} JOUEUR;

int NBJOUEUR = 4;

JOUEUR joueur[] = {
    {"PIER",10,10,2},
    {"CATH",11,10,3},
    {"PAUL",12,10,5},
    {"JEAN",13,10,6}};

int COULEUR_FOND = 7; // indice paire de couleur pour le fond ecran

int
main(int argc, char *argv[])
{
    int num = 0;
    int i;
    int numJoueur;
    short r;
    char rep[3];
    JOUEUR jC;
    short v;
    WINDOW *wjeu, *wmes; // fenetre deplacement et fenetre message

    set_ch_vid(rep);

    int l,c;

    /* initialize your non-curses data structures here */

    (void) signal(SIGINT, finish); // arrange interrupts to terminate */

    deb_vac();

    keypad(stdscr, TRUE);

```

```

if (has_colors())
{
    start_color();

    /*
    * Simple color assignment, often all we need. Color pair 0 cannot
    * be redefined. This example uses the same value for the color
    * pair as for the foreground color, though of course that is not
    * necessary:
    */
    init_pair(1, COLOR_RED,      COLOR_YELLOW);
    init_pair(2, COLOR_GREEN,    COLOR_YELLOW);
    init_pair(3, COLOR_YELLOW,   COLOR_RED);
    init_pair(4, COLOR_BLUE,     COLOR_YELLOW);
    init_pair(5, COLOR_CYAN,     COLOR_BLUE);
    init_pair(6, COLOR_MAGENTA,  COLOR_YELLOW);
    init_pair(7, COLOR_BLACK,    COLOR_BLACK); //fond
}

// creation du cadre du jeu

wjeu = newwin(LINES-10, COLS-5, 1, 3);    /* creation fenetre deplacement */
wmes = newwin(5, COLS-5, LINES-9, 3);    // creation fenetre message
box(wjeu, ACS_VLINE, ACS_HLINE);
box(wmes, ACS_VLINE, ACS_HLINE);
wrefresh(wjeu);
wrefresh(wmes);

keypad(wmes, TRUE);

// affichage initial des joueurs

for( i=0; i< NBJOUEUR; i++)
{
    wattrset(wjeu, COLOR_PAIR(joueur[i].couleur));
    mvwaddstr(wjeu, joueur[i].l, joueur[i].c, joueur[i].nom);
}

wrefresh(wjeu);

numJoueur = 0;

jC = joueur[numJoueur]; // joueur courant

for (;;)
{
    wattrset(wjeu, COLOR_PAIR(jC.couleur));
    jC = joueur[numJoueur]; // joueur courant
    wcri_ent(wmes, 3, 1, jC.l, "ci");
    mvwaddstr(wjeu, jC.l, jC.c, jC.nom);

    wrefresh(wjeu);
}

```

```

r= wlire_curs(stdscr,0,0,rep);

if ( r == 0 || r == 1 ) // fleche ou tf
{

    switch(rep[0])
    {
        case FLD :
            wattrset(wjeu,COLOR_PAIR(COULEUR_FOND));
            mvwaddstr(wjeu,jC.l, jC.c, "      ");
            jC.c++;
            joueur[numJoueur] = jC;
            break;

        case FLG :
            wattrset(wjeu,COLOR_PAIR(COULEUR_FOND));
            mvwaddstr(wjeu,jC.l, jC.c, "      ");
            jC.c--;
            joueur[numJoueur] = jC;
            break;

        case FLB :
            wattrset(wjeu,COLOR_PAIR(COULEUR_FOND));
            mvwaddstr(wjeu,jC.l, jC.c, "      ");
            jC.l++;
            joueur[numJoueur] = jC;
            break;

        case FLH :
            wattrset(wjeu,COLOR_PAIR(COULEUR_FOND));
            mvwaddstr(wjeu,jC.l, jC.c, "      ");
            jC.l--;
            joueur[numJoueur] = jC;
            break;

        case 'A' : // touche F1 on bascule
                    // changement de pion
            numJoueur =(numJoueur + 1) % NBJOUEUR;
            break;
    }
}

}
fin_vac();

}

static void finish(int sig)
{
    fin_vac();

    exit(0);
}

```

