

(*Modules: ouverture des modules nécessaires***)**

```
#open "graphics";  
#open "random";
```

(*Variables: déclaration des constantes et des types***)**

(*Type des cases*)

```
type case = Noir | Blanc | Vide;;
```

(*Couleurs du plateau*)

(*couleur des cadres des cases du tableau*)(*noir*)

```
let coul_cadre=rgb 0 0 0;;
```

(*couleur du fond des cases du tableau*)(*vert foncé*)

```
let coul_fond=rgb 0 128 0;;
```

(*couleur des reflets des pions*)(*blanc*)

```
let coul_reflet= rgb 255 255 255;;
```

(*couleur du sur lignage des cases*)(*vert clair*)

```
let coul_surlignage= rgb 0 170 0;;
```

(*couleur de sélection des cases*)(*vert clair+*)

```
let coul_select= rgb 0 210 0;;
```

(*couleur de prévisualisation des coups*)(*vert clair+*)

```
let coul_coup= rgb 0 0 128;;
```

(*couleur des pions*) (*blanc-gris*)(*noir*)

```
let coul_blanc=rgb 223 218 207;;
```

```
let coul_noir=rgb 0 0 0;;
```

(*Coordonnées de référence*)

```
let x0,y0=30,30;;
```

(*Dimension de référence*)

```
let dim_cote=60;;
```

```
let dim_rayon=(4*dim_cote)/(2*5) ;;
```

```
let dim_nbcote=8;;
```

(*Plateau de jeu*)

```
let Plateau = make_matrix dim_nbcote dim_nbcote Vide;;
```

```
Plateau.(3).(3)<-Noir ;;
```

```
Plateau.(4).(4)<-Noir ;;
```

```
Plateau.(3).(4)<-Blanc ;;
```

```
Plateau.(4).(3)<-Blanc ;;
```

(*Fonctions de représentations du plateau***)**

(*Dessin des cases du plateau*)

```
let fait_case x y coul_cadre coul_fond=
```

```
set_color coul_cadre;
```

```
moveto x y;
```

```
lineto (x+dim_cote) y;
```

```
lineto (x+dim_cote) (y+dim_cote);
```

```
lineto x (y+dim_cote);
```

```
lineto x y;
```

```

set_color coul_fond;
fill_rect (x+1) (y+1) (dim_cote-2) (dim_cote-2);
(*Dessin des pions*)
let fait_pion x y coul_cadre coul_fond coul_pion =
fait_case x y coul_cadre coul_fond ;
set_color coul_pion ;
fill_circle (x+ dim_cote/2) (y+dim_cote/2) dim_rayon ;
set_color coul_reflet ;
draw_circle (x+ dim_cote/2) (y+dim_cote/2) (dim_rayon-2) ;;
(*Affichage de la case (i,j) *)
let affiche_case plateau i j=
let x,y=((i*dim_cote+x0),(j*dim_cote+y0)) in
if plateau.(i).(j)=Vide then fait_case x y coul_cadre coul_fond
else if plateau.(i).(j)=Noir then fait_pion x y coul_cadre coul_fond coul_noir
else if plateau.(i).(j)=Blanc then fait_pion x y coul_cadre coul_fond coul_blanc ;;
(*Affichage du plateau entier*)
let affiche_plateau plateau=
for i=0 to (vect_length plateau-1) do
for j=0 to (vect_length plateau.(0)-1) do
affiche_case plateau i j;
done;
done;;
(*Affichage des coups possibles*)
let rec affiche_coup = function
| []->()
| (i,j) ::l->fait_case (x0+dim_cote*i) (y0+dim_cote*j) coul_cadre coul_coup ;
affiche_coup l ;;
(*Effacement des coups possibles*)
let rec affiche_pas_coup = function
| []->()
| (i,j) ::l->fait_case (x0+dim_cote*i) (y0+dim_cote*j) coul_cadre coul_fond ;
affiche_pas_coup l ;;
(*Fonctions de manipulation du plateau*****
(*Possibilité d'une ligne*)
let ligne plateau coul1 i j k l=
let coul2=ref Noir in if coul1=Noir then coul2:=Blanc;
let bool=ref true in
if k>7 || l>7 || k<0 || l<0 then bool:=false else
if plateau.(k).(l) <> !coul2 then bool:=false else
begin
let rec aux = fun
| 7 j2->false
| i2 7->false
| i2 j2->if plateau.(i2).(j2)=coul1 then true else
if plateau.(i2).(j2)=Vide then false else

```

```

                                aux (i2+k-i) (j2+l-j) in

bool:= aux k l;
end;
!bool;;
(*Possibilité d'un coup en (i,j)*)
let possible plateau coul1 i j=
let bool = ref false in
if plateau.(i).(j)=Vide then
begin
for k= -1 to 1 do
for l= -1 to 1 do
    bool:= !bool || (ligne plateau coul1 i j (i+k) (j+l));
done; done;end ; !bool;;
(*Evaluation des coups possibles*)
let coup plateau coul1=
let rec aux = fun
| 7 7-> if (possible plateau coul1 7 7) then [(7,7)] else []
| i 7-> if (possible plateau coul1 i 7) then (i,7)::(aux (i+1) 0) else (aux (i+1) 0)
| i j -> if (possible plateau coul1 i j) then (i,j)::(aux i (j+1)) else (aux i (j+1))    in
aux 0 0;;
(*Cliquer d'une case du plateau*)
let cliquer plateau coul1 list=
    let case=ref (0,0) in
    let mauvais_clique=ref true in
while !mauvais_clique do
    let attend=ref (wait_next_event[Button_down;Mouse_motion;Key_pressed]) in
    let x,y=(!attend.mouse_x -x0),(!attend.mouse_y -y0) in
(*Si le curseur est dans le plateau*)
if    x<(dim_cote*(vect_length plateau))
    && y<(dim_cote*(vect_length plateau.(0)))
    && x>=0 && y>=0
then
    begin
    (*On repeint la case précédente*)
    if plateau.(fst !case).(snd !case)=Vide then
        fait_case ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_fond
    else if plateau.(fst !case).(snd !case)=Noir then
        fait_pion ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_fond
coul_noir
    else if plateau.(fst !case).(snd !case)=Blanc then
        fait_pion ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_fond
coul_blanc;
    (*On récupère la case que l'on pointe*)
    case:=((x quo dim_cote), (y quo dim_cote));
    if plateau.(fst !case).(snd !case)=Vide then

```

```

    fait_case ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_surlignage
else if plateau.(fst !case).(snd !case)=Noir then
    fait_pion ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_surlignage
coul_noir
else if plateau.(fst !case).(snd !case)=Blanc then
    fait_pion ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_surlignage
coul_blanc;
(*Si la case est vide*)
if plateau.(fst !case).(snd !case)=Vide then
    begin
        (*Si on appuie sur une touche*)
        if !attend.keypressed then begin (affiche_coup list); let a=wait_next_event[Key_pressed] in
(affiche_pas_coup list); end
        (*Si on clique sur une valide*)
        else if !attend.button && plateau.(fst !case).(snd !case)=Vide then
            begin
                fait_case ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre coul_select;
                let attend2=ref (wait_next_event[Button_up;]) in
                let x2,y2=(!attend2.mouse_x -x0),(!attend2.mouse_y -y0) in
                let case2=((x2 quo dim_cote), (y2 quo dim_cote)) in
                if case2 = !case then
                    begin
                        sound 120 10;
                        mauvais_clique:=false;
                    end;
                end
            else fait_case ((fst !case)*dim_cote+x0) ((snd !case)*dim_cote+y0) coul_cadre
coul_surlignage;
            end;
        end
done;
(fst !case,snd !case);;

```

```

(******Fonction du jeu*)

```

```

let test ()=
open_graph "800x600+200+100";
affiche_plateau Plateau ;
cliquer Plateau Blanc (coup Plateau Blanc) ;
close_graph();;

```

```

test() ;;

```