

(*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_arc (x+ dim_cote/2) (y+dim_cote/2) (dim_rayon-3) (dim_rayon-3) 90 180 ;;
(*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*****
(*Élément dans une liste*)
let rec dans = fun
| a []-> false
| a (b::l)->a=b || (dans a l) ;;
(*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

```

```

|8 j2-> false
|i2 8->false
|(-1) j2-> false
|i2 (-1)->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;;
(*Calcul du score*)
let score plateau=
let b,n=ref 0,ref 0 in
for i=0 to 7 do
for j=0 to 7 do
    if plateau.(i).(j)=Noir then incr n else if plateau.(i).(j)=Blanc then incr b;
done; done;
(!b,!n);
(*Affichage du score*)
let affiche_score plateau=
let b,n=score plateau in
set_color coul_blanc;
fill_rect (x0+dim_nbcas*dim_cote+3) (y0+dim_nbcas*dim_cote-20) 100 20;
set_color coul_noir;
fill_rect (x0+dim_nbcas*dim_cote+3) (y0+dim_nbcas*dim_cote-45) 100 20;
set_color coul_noir;
moveto (x0+dim_nbcas*dim_cote+10) (y0+dim_nbcas*dim_cote- 20);
draw_string ((string_of_int b)^" blanc(s)");
moveto (x0+dim_nbcas*dim_cote+10) (y0+dim_nbcas*dim_cote-45);

```

```

draw_string ((string_of_int n)^" noir(s) ");;
(*Affichage du tour*)
let affiche_tour coul=
let couleur=ref coul_blanc in
if coul=Noir then couleur := coul_noir ;
set_color coul_noir;
moveto (x0+dim_nbcase*dim_cote+10) (y0+dim_nbcase*dim_cote- 70);
draw_string ("C'est aux »);
set_color !couleur;
fill_rect (x0+dim_nbcase*dim_cote+90) (y0+dim_nbcase*dim_cote- 69) 15 15;;
(*Application d'une ligne *)
let fait_ligne coul1 coul2 i j k l=
if (ligne Plateau coul1 i j k l) then begin
let i2,j2=ref k, ref l in
Plateau.(i).(j)<-coul1;
while Plateau.(!i2).(!j2)=coul2 do
Plateau.(!i2).(!j2)<-coul1;
i2:= !i2 +k-i; j2:= !j2 +l-j;
done; end;;
(*Application d'un coup*)
let fait_coup plateau coul1 i j=
let coul2=ref Noir in if coul1=Noir then coul2:=Blanc;
for k= -1 to 1 do
for l= -1 to 1 do
fait_ligne coul1 (!coul2) i j (i+k) (j+l);
done; done;;
(*Vérification de victoire*)
let victoire plateau =
let gagnant=ref (false,Vide) in
let b,n=score plateau in
if ((coup plateau Noir),(coup plateau Blanc))=([],[]) then begin
if b>n then gagnant:=(true,Blanc)
else if b<n then gagnant:=(true,Noir)
else gagnant:=(true,Vide);
end ;
!gagnant ;;
(*Clique 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;
        begin
            (*Si on appuie sur une touche*)
            if !attend.keypressed then begin
                if !attend.key <> `027` then begin
                    affiche_case plateau (fst !case) (snd !case);
                    (affiche_coup list);
                    let a=wait_next_event[Key_pressed] in
                    (affiche_pas_coup list);
                    end
                else begin
                    close_graph();
                    failwith « Merci d'avoir joué » ;
                    end;
                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;
  if (dans !case list) then begin
    fait_coup plateau coul1 (fst !case) (snd !case) ;
    mauvais_clique:=false;
    end ;
  end;
end;
end;
end else affiche_case plateau (fst !case) (snd !case);
done;;

```

(*****Fonction du jeu*)

```

let test ()=
  open_graph "";
  close_graph ();
  open_graph "800x600+200+100";
  let tour=ref Noir in
  while not (fst (victoire Plateau)) do
    if !tour=Noir then tour:=Blanc else tour:=Noir;
    if (coup Plateau !tour)=[] then if !tour=Noir then tour:=Blanc else tour:=Noir;
    affiche_tour !tour ;
    affiche_plateau Plateau ;
    affiche_score Plateau;
    cliquer Plateau !tour (coup Plateau !tour) ;
  done ;
  close_graph();;

  test() ;;

```