**(\*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\_nbcase=8;;

**(\*Plateau de jeu\*)**

let Plateau = make\_matrix dim\_nbcase dim\_nbcase 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;;

**(\*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;  
 **(\*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() ;;