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

#open "graphics";;

#open "random";;

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

type sondage = Explosion | Bord | Voisin of int;;

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

let coul\_cadre=rgb 0 0 0;; (\*noir\*)

(\*couleur du fond des cases du tableau\*)

let coul\_fond=rgb 192 192 192;; (\*gris\*)

(\*couleur des reflets des cases du tableau\*)

let coul\_reflet\_inf= rgb 128 128 128;; (\*gris foncé\*)

let coul\_reflet\_sup= rgb 255 255 255;; (\*blanc\*)

let coul\_select=rgb 225 225 225;; (\*gris clair\*)

(\*couleur des mines\*)

let coul\_mine=rgb 0 0 0;; (\*noir\*)

(\*couleur des drapeaux\*)

let coul\_support=rgb 0 0 0;; (\*noir\*)

let coul\_drapeau=rgb 255 0 0;; (\*rouge\*)

(\*couleur faux\*)

let coul\_non=rgb 255 0 0;; (\*rouge\*)

(\*couleur vrai\*)

let coul\_oui=rgb 0 128 0;; (\*vert foncé\*)

(\*couleurs des numéros dans les cases du tableau\*)

let coul= function

|1->rgb 0 0 255 (\*bleu\*)

|2->rgb 0 128 0 (\*vert foncé\*)

|3->rgb 255 0 0 (\*rouge\*)

|4->rgb 0 0 128 (\*bleu foncé\*)

|5->rgb 128 0 0 (\*rouge foncé\*)

|6->rgb 255 255 0 (\*jaune\*)

|7->rgb 255 0 255 (\*violet\*)

|8->rgb 0 0 0 (\*noir\*)

|\_->white;;

(\*Coin superieur gauche du tableau\*)

let x0,y0=(10,420);;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Le champ est une matrice d'int avec le codage suivant:

0 : champ sans mine voisine (dévoilé)

n : champ à n mines voisines (dévoilé)

9 : champ libre (caché)

10 : champ miné (caché)

11 : champ miné (dévoilé, pour la fin)

13 : bord (non affiché)

14 : mine activé (dévoilé, pour la fin)

15 : victoire (non affiché)

16 : drapeau vrai (caché)

160: drapeau vrai (dévoilé, pour la fin)

17 : drapeau faux (caché)

170: drapeau faux (dévoilé, pour la fin)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fonctions graphiques et d'affichage\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dessin des cases\*)

let fait\_carre x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf=

set\_color coul\_cadre;

moveto x y;

lineto (x+25) y;

lineto (x+25) (y-25);

lineto x (y-25);

lineto x y;

set\_color coul\_fond;

fill\_rect (x+1) (y-24) 24 24;

set\_color coul\_reflet\_sup;

fill\_rect (x+1) (y-24) 3 24;

fill\_rect (x+1) (y-3) 24 3;

set\_color coul\_reflet\_inf;

fill\_rect (x+4) (y-24) 21 3;

fill\_rect (x+22) (y-24) 3 21;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dessin des mines\*)

let fait\_mine x y coul\_cadre coul\_fond coul\_mine coul\_reflet\_sup=

fait\_carre x y coul\_cadre coul\_fond coul\_fond coul\_fond;

set\_color coul\_mine;

fill\_circle (x+13) (y-13) 8;

fill\_rect (x+2) (y-13) 22 3;

fill\_rect (x+12) (y-23) 3 22;

set\_color coul\_reflet\_sup;

fill\_circle (x+11) (y-11) 2;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dessin des drapeau\*)

let fait\_drapeau x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf coul\_drapeau coul\_support=

fait\_carre x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf;

set\_color coul\_support;

fill\_rect (x+5) (y-20) 17 3;

fill\_rect (x+16)(y-20) 3 15;

set\_color coul\_drapeau;

(\*fill\_poly bug dès qu'on bouge l'écran vers la droite...mais tant pis, pas envie de dessiner les triangles à la main...\*)

fill\_poly [|((x+5),(y-11));((x+16),(y-7));((x+16),(y-15))|];;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Affichage de la case (i,j) \*)

let affiche\_case champ i j=

let x,y=((i\*25+x0),(size\_y() -j\*25-y0)) in

if champ.(i).(j)= 9 then fait\_carre x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf

else if champ.(i).(j)=10 then fait\_carre x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf

else if champ.(i).(j)=11 then fait\_mine x y coul\_cadre coul\_fond coul\_mine coul\_reflet\_sup

else if champ.(i).(j)=14 then fait\_mine x y coul\_cadre coul\_non coul\_mine coul\_reflet\_sup

else if champ.(i).(j)=16 then fait\_drapeau x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf coul\_drapeau coul\_support

else if champ.(i).(j)=160 then fait\_mine x y coul\_cadre coul\_oui coul\_mine coul\_reflet\_sup

else if champ.(i).(j)=17 then fait\_drapeau x y coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf coul\_drapeau coul\_support

else if champ.(i).(j)=170 then fait\_carre x y coul\_cadre coul\_non coul\_non coul\_non

else (\*ie à n voisin\*) begin fait\_carre x y coul\_reflet\_inf white white white;

set\_color (coul (champ.(i).(j)));

moveto (x+8) (y-23);

draw\_string (string\_of\_int champ.(i).(j))

end;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Affichage du champ entier \*)

let affiche\_champ champ=

for i=1 to (vect\_length champ -2) do

for j=1 to (vect\_length champ.(0)-2) do

affiche\_case champ i j;

done;

done;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Affichage du nombre de mines restant (+vérification de victoire) \*)

let affiche\_score champ nb\_mines=

let nb\_drapeau=ref 0 in

let nb\_case=ref 0 in

for i=1 to (vect\_length champ -2) do

for j=1 to (vect\_length champ.(0)-2) do

if champ.(i).(j)=16 || champ.(i).(j)=17 then incr nb\_drapeau;

if champ.(i).(j)=9 || champ.(i).(j)=10 then incr nb\_case;

if champ.(i).(j)=14 then nb\_case:=1000;

done;

done;

(\*afichage\*)

set\_color coul\_non;

moveto (x0+50) (size\_y()-y0-5);

draw\_string "Mine(s) restante(s): ";

moveto (x0+210) (size\_y()-y0-5);

draw\_string (string\_of\_int (nb\_mines - !nb\_drapeau));

(\*victoire\*)

if !nb\_drapeau + !nb\_case = nb\_mines then champ.(0).(0)<-15;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fonctions d'utilisation du champ\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Création d'un champ \*)

let creer\_champ n p nb\_mine=

(\*On créer tout libre\*)

let champ=make\_matrix (n+2) (p+2) 9 in

(\*On marque les bords\*)

for k=0 to n+1

do champ.(k).(0)<-13;champ.(k).(p+1)<-13 done;

for k=0 to p+1

do champ.(0).(k)<-13;champ.(n+1).(k)<-13 done;

(\*On place les bombes\*)

let k=ref 0 in

while !k<>nb\_mine do

let i,j=((random\_\_int n)+1),((random\_\_int p)+1) in

if not champ.(i).(j)=10 then begin champ.(i).(j)<-10;incr k;end;

done;

champ;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sondage d'une case (i,j) \*)

let sonde champ i j =

if champ.(i).(j)=10 then Explosion

else if champ.(i).(j)=13 then Bord

else begin

let v=ref 0 in

for k=(-1) to 1 do

for l=(-1) to 1 do

if champ.(i+k).(j+l)=10 || champ.(i+k).(j+l)=11 || champ.(i+k).(j+l)=16 then incr v;

done; done;Voisin(!v);

end;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Conséquence du sondage en (i,j) \*)

let rec actualise champ i j (\*sondage\*)= function

|Explosion->champ.(0).(0)<-14;

champ.(i).(j)<-14;

|Bord->();

|Voisin(0)->champ.(i).(j)<-0; (\*on sonde toutes les cases mitoyennes sous reserve qu'elles soit sondables\*)

if champ.(i+1).(j-1)=10 || champ.(i+1).(j-1)=9

then actualise champ (i+1) (j-1) (sonde champ (i+1) (j-1));

if champ.(i+1).(j+0)=10 || champ.(i+1).(j+0)=9

then actualise champ (i+1) (j+0) (sonde champ (i+1) (j+0));

if champ.(i+1).(j+1)=10 || champ.(i+1).(j+1)=9

then actualise champ (i+1) (j+1) (sonde champ (i+1) (j+1));

if champ.(i+0).(j-1)=10 || champ.(i+0).(j-1)=9

then actualise champ (i+0) (j-1) (sonde champ (i+0) (j-1));

if champ.(i+0).(j+1)=10 || champ.(i+0).(j+1)=9

then actualise champ (i+0) (j+1) (sonde champ (i+0) (j+1));

if champ.(i-1).(j-1)=10 || champ.(i-1).(j-1)=9

then actualise champ (i-1) (j-1) (sonde champ (i-1) (j-1));

if champ.(i-1).(j+0)=10 || champ.(i-1).(j+0)=9

then actualise champ (i-1) (j+0) (sonde champ (i-1) (j+0));

if champ.(i-1).(j+1)=10 || champ.(i-1).(j+1)=9

then actualise champ (i-1) (j+1) (sonde champ (i-1) (j+1));

|Voisin(n)->champ.(i).(j)<-n;;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fonctions de jeu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Gestion du gameplay\*)

let cliquer champ nb\_mine=

let carre=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

if x<25\*(vect\_length champ-1) && y>(size\_y()-25\*(vect\_length champ.(0)-1)) && x>=0 && y<=size\_y() then

begin

if (champ.(fst !carre).(snd !carre)=10 || champ.(fst !carre).(snd !carre)=9) then

fait\_carre ((fst !carre)\*25+x0) (size\_y() -(snd !carre)\*25-y0) coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf;

carre:=((x quo 25), ((size\_y()-y) quo 25));

if (champ.(fst !carre).(snd !carre)=10 || champ.(fst !carre).(snd !carre)=9) ||

(champ.(fst !carre).(snd !carre)=16 || champ.(fst !carre).(snd !carre)=17) then

begin

if !attend.keypressed then

begin

if champ.(fst !carre).(snd !carre)=9 then champ.(fst !carre).(snd !carre)<-17

else if champ.(fst !carre).(snd !carre)=10 then champ.(fst !carre).(snd !carre)<-16

else if champ.(fst !carre).(snd !carre)=16 then champ.(fst !carre).(snd !carre)<-10

else if champ.(fst !carre).(snd !carre)=17 then champ.(fst !carre).(snd !carre)<-9;

affiche\_case champ (fst !carre) (snd !carre);

affiche\_score champ nb\_mine;

end

else if !attend.button && (champ.(fst !carre).(snd !carre)=9 || champ.(fst !carre).(snd !carre)=10) then

begin

fait\_carre ((fst !carre)\*25+x0) (size\_y() -(snd !carre)\*25-y0) coul\_cadre coul\_reflet\_inf coul\_reflet\_inf coul\_reflet\_inf;

let attend2=ref (wait\_next\_event[Button\_up;]) in

let x2,y2=(!attend2.mouse\_x -x0),(!attend2.mouse\_y +y0) in

let carre2=((x2 quo 25), ((size\_y()-y2) quo 25)) in

if carre2 = !carre then

begin

sound 1000 10;

mauvais\_clique:=false;

end;

end

else if (champ.(fst !carre).(snd !carre)=10 || champ.(fst !carre).(snd !carre)=9) then

fait\_carre ((fst !carre)\*25+x0) (size\_y() -(snd !carre)\*25-y0) coul\_cadre coul\_reflet\_sup coul\_select coul\_reflet\_inf;

end;

end

else if (champ.(fst !carre).(snd !carre)=10 || champ.(fst !carre).(snd !carre)=9) then

fait\_carre ((fst !carre)\*25+x0) (size\_y() -(snd !carre)\*25-y0) coul\_cadre coul\_reflet\_sup coul\_fond coul\_reflet\_inf;

done;

(fst !carre,snd !carre);;

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

let deminer n p nb\_mines=

let champ=creer\_champ n p nb\_mines in

open\_graph "800x600+200+100";

affiche\_champ champ;

affiche\_score champ nb\_mines;

(\*jeu\*)

while champ.(0).(0)=13 do

let (i,j)=cliquer champ nb\_mines in actualise champ i j (sonde champ i j);

affiche\_champ champ;

affiche\_score champ nb\_mines;

done;

(\*affichage de fin\*)

for k=1 to (vect\_length champ-2) do

for l=1 to (vect\_length champ.(0)-2) do

if champ.(k).(l)=10 then champ.(k).(l)<-11;

if champ.(k).(l)=16 then champ.(k).(l)<-160;

if champ.(k).(l)=17 then champ.(k).(l)<-170;

done;

done;

affiche\_champ champ;

if champ.(0).(0)=14 then

begin

moveto (x0+30) (size\_y()-y0+10);

set\_color red;

draw\_string "Vous avez perdu !! (cliquez pour quitter)";

sound 80 100;

sound 70 100;

sound 60 100;

sound 40 300;

end;

if champ.(0).(0)=15 then

begin

moveto (x0+30) (size\_y()-y0+10);

set\_color green;

draw\_string "Vous avez gagné !! (cliquez pour quitter)";

sound 60 100;

sound 70 100;

sound 80 100;

sound 100 300;

end;

let a=wait\_next\_event[Button\_down] in ();

close\_graph();;

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Lancement du jeu\*)

deminer 15 12 10;;