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

#open "graphics";;

#open "sys";;

#open "random";;

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

let gris= rgb 192 192 192;;

let grisfonce= rgb 128 128 128;;

let grisclaire=rgb 225 225 225;;

let coul= function

|1->rgb 0 0 255

|2->rgb 0 128 0

|3->rgb 255 0 0

|4->rgb 0 0 128

|5->rgb 128 0 0

|6->rgb 255 255 0

|7->rgb 255 0 255

|8->rgb 0 0 0

|\_->white;;

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

(\* 9: champ libre caché

10: champ miné caché

11: champ miné dévoilé

13: bord

14: mine activé

15: victoire

16: drapeau vrai

17: drapeau faux

0: champ sans mine voisine dévoilé

n: champ à n mines voisines dévoilé \*)

let creer\_champ n p nb=

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

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;

let k=ref 0 in

while !k<>nb 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;;

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;;

let fait\_carre x y col0 col1 col2 col3=

set\_color col0;

moveto x y;

lineto (x+25) y;

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

lineto x (y-25);

lineto x y;

set\_color col2;

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

set\_color col1;

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

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

set\_color col3;

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

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

let fait\_bombe x y col0 col1 col2 col3=

set\_color col0;

moveto x y;

lineto (x+25) y;

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

lineto x (y-25);

lineto x y;

set\_color col1;

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

set\_color col2;

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 col3;

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

let fait\_drapeau x y col0 col1 col2 col3 col4 col5=

fait\_carre x y col0 col1 col2 col3;

set\_color col5;

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

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

set\_color col4;

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

let cliquer champ nb\_mines=

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 -50),(!attend.mouse\_y +25) 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+50) (size\_y() -(snd !carre)\*25-25) black white gris grisfonce;

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\_mines;

end

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

begin

fait\_carre ((fst !carre)\*25+50) (size\_y() -(snd !carre)\*25-25) black grisfonce grisfonce grisfonce;

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

let x2,y2=(!attend2.mouse\_x -50),(!attend2.mouse\_y +25) in

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

if carre2 = !carre then

begin

sound 1000 10;

mauvais\_clique:=false;

fait\_carre ((fst !carre)\*25+50) (size\_y() -(snd !carre)\*25-25) grisfonce gris gris gris;

end;

end

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

fait\_carre ((fst !carre)\*25+50) (size\_y() -(snd !carre)\*25-25) black white grisclaire grisfonce;

end;

end

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

fait\_carre ((fst !carre)\*25+50) (size\_y() -(snd !carre)\*25-25) black white gris grisfonce

done;(fst !carre,snd !carre);;

let affiche\_case champ i j=

if champ.(i).(j)=10 || champ.(i).(j)=9 then

fait\_carre (i\*25+50) (size\_y() -j\*25-25) black white gris grisfonce

else if champ.(i).(j)=11 then fait\_bombe (i\*25+50) (size\_y() -j\*25-25) black gris black white

else if champ.(i).(j)=14 then fait\_bombe (i\*25+50) (size\_y() -j\*25-25) black red black white

else if champ.(i).(j)=16 then fait\_drapeau (i\*25+50) (size\_y() -j\*25-25) black white gris grisfonce red black

else if champ.(i).(j)=17 then fait\_drapeau (i\*25+50) (size\_y() -j\*25-25) black white gris grisfonce red black

else if champ.(i).(j)=160 then fait\_bombe (i\*25+50) (size\_y() -j\*25-25) black green black white

else if champ.(i).(j)=170 then fait\_carre (i\*25+50) (size\_y() -j\*25-25) black red red red

else

begin

fait\_carre (i\*25+50) (size\_y() -j\*25-25) grisfonce white white white;

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

moveto (i\*25+58) (size\_y() -j\*25-48);

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

end;;

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;;

let rec actualise champ i j= function

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

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

|Bord->();

|Voisin(0)->champ.(i).(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+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;;

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;

done;

done;

set\_color red;

moveto 100 (size\_y()-50);

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

moveto 270 (size\_y()-50);

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

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

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;

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;

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 80 (size\_y()-20);

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 80 (size\_y()-20);

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();;

deminer 15 15 20;;