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

#open "graphics";;

#open "sys";;

#open "random";;

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

let x\_ecran,y\_ecran=25,25;;

let lo\_ecran=700;;

let ha\_ecran=400;;

let st\_ecran="800x500+100+100";;

let r\_balle= ref 6;;

let nb\_bloc\_lo=(int 15)+1;;

let nb\_bloc\_ha=(int 15)+1;;

let nb\_bloc = nb\_bloc\_lo\*nb\_bloc\_ha;;

let lo\_bloc=37;;

let ha\_bloc=15;;

let xi\_bloc,yi\_bloc=(lo\_ecran-nb\_bloc\_lo\*(lo\_bloc+1))/2+x\_ecran,(ha\_ecran-nb\_bloc\_ha\*(ha\_bloc+1))+y\_ecran-50;;

let lo\_barre= ref 70;;

let ha\_barre=10;;

let y\_barre=y\_ecran-ha\_barre;;

let pas\_barre=50;;

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

let vies=ref 3;;

let x\_barre=ref (x\_ecran + lo\_ecran/2 - !lo\_barre/2);;

let x\_balle,y\_balle=ref 100,ref 100;;

let pax\_balle,pay\_balle=ref 1,ref 1;;

let continuer = ref true;;

(\*\*\*\*\*\*\*Variables: initialisation des blocs\*\*\*\*\*\*\*)

let bloc= (make\_vect nb\_bloc (0,0));;

for l=0 to (nb\_bloc\_ha-1) do

for k=0 to (nb\_bloc\_lo-1) do

bloc.(nb\_bloc\_lo\*l+k)<-(xi\_bloc+k\*(lo\_bloc+1),yi\_bloc+l\*(ha\_bloc+1));

done;

done;;

(\*\*\*\*\*\*\*Fonction: extraction d'un composant d'un couple\*\*\*\*\*\*\*)

let extract = fun

|(x,y) 1->x

|(x,y) \_->y;;

(\*\*\*\*\*\*\*Fonction: traçage d'un rectangle lo\*ha au point (x,y)\*\*\*\*\*\*\*)

let rect x y lo ha =

moveto x y;

lineto x (y+ha-1);

lineto (x+lo-1) (y+ha-1);

lineto (x+lo-1) y;

lineto x y;;

(\*\*\*\*\*\*\*Fonction: affichage de la balle\*\*\*\*\*\*\*)

let draw\_balle c=

set\_color c;

fill\_circle !x\_balle !y\_balle !r\_balle;

set\_color white;

fill\_circle (!x\_balle+ !r\_balle/2) (!y\_balle+ !r\_balle/2) (!r\_balle/4);;

(\*\*\*\*\*\*\*Fonction: affichage de la barre\*\*\*\*\*\*\*)

let draw\_barre () =

if !x\_barre <> extract (mouse\_pos()) 1 then

begin set\_color white;

fill\_rect (!x\_barre-ha\_barre) y\_barre (!lo\_barre+2\*ha\_barre) (ha\_barre);

x\_barre:=extract (mouse\_pos()) 1;

if x\_ecran + lo\_ecran < !x\_barre+ !lo\_barre then x\_barre:= x\_ecran + lo\_ecran- !lo\_barre;

if x\_ecran > !x\_barre then x\_barre:= x\_ecran;

set\_color green;

rect !x\_barre y\_barre !lo\_barre ha\_barre ;

fill\_rect !x\_barre y\_barre ( !lo\_barre-2) (ha\_barre-2);

fill\_circle !x\_barre (y\_barre+(ha\_barre-2)/2) (ha\_barre/2);

fill\_circle (!x\_barre+ !lo\_barre) (y\_barre+(ha\_barre-2)/2) (ha\_barre/2);

end;;

(\*\*\*\*\*\*\*Fonction: affichage des vie\*\*\*\*\*\*\*)

let vie () =

set\_color red;

moveto (x\_ecran+lo\_ecran+5) (y\_ecran+ha\_ecran-20);

draw\_string ("x"^(string\_of\_int !vies));;

(\*\*\*\*\*\*\*Fonction: suppression d'un bloc\*\*\*\*\*\*)

let efface n =

set\_color white;

fill\_rect ((extract bloc.(n) 1)) ((extract bloc.(n) 2)) (lo\_bloc+1) (ha\_bloc+1);

bloc.(n)<-(0,0);;

(\*\*\*\*\*\*\*Fonction: initialisation de l'écran\*\*\*\*\*\*\*)

let init () =

rect x\_ecran y\_ecran lo\_ecran ha\_ecran;

set\_color white;

moveto x\_ecran y\_ecran;

lineto (x\_ecran+lo\_ecran) y\_ecran;

for k=0 to (nb\_bloc-1) do

set\_color black;

rect (extract bloc.(k) 1) (extract bloc.(k) 2) lo\_bloc ha\_bloc;

set\_color red;

fill\_rect ((extract bloc.(k) 1)+1) ((extract bloc.(k) 2)+1) (lo\_bloc-3) (ha\_bloc-3);

vie();

draw\_barre();

done;;

(\*\*\*\*\*\*\*Fonction: pause\*\*\*\*\*\*\*)

let pause () =

moveto (x\_ecran+lo\_ecran/2)(y\_ecran+ha\_ecran);

set\_color red; draw\_string("PAUSE");

while not key\_pressed() || (wait\_next\_event [Key\_pressed]).key <> `p` do () done;

moveto (x\_ecran+lo\_ecran/2)(y\_ecran+ha\_ecran);

set\_color white; draw\_string("PAUSE");;

(\*\*\*\*\*\*\*Fonction: création de bonus aléatoires\*\*\*\*\*\*\*)

let hasard () =

sound 120 150;

match (int 3) with

|0->begin

vies:=!vies + (int 3) -1;

vie();

end

|1->begin

draw\_balle white;

if int(2)=0 then r\_balle:= !r\_balle\*2 else r\_balle:= !r\_balle/2;

draw\_balle blue;

end;

|2->begin

if int(2)=0 then lo\_barre:= 200 else lo\_barre:= 15;

fill\_rect x\_ecran (y\_ecran-ha\_barre) lo\_ecran ha\_barre;

decr x\_barre;

draw\_barre();

end

|\_->();;

(\*\*\*\*\*\*\*Fonction: gestion des événements\*\*\*\*\*\*\*)

let prise\_event () =

if key\_pressed() then

match (wait\_next\_event [Key\_pressed;Poll]).key with

|``->vies:=-1

|`p`->pause()

|`\013`->hasard()

|\_->();;

(\*\*\*\*\*\*\*Fonction: attente durant t (float) seconde(s)\*\*\*\*\*\*)

let wait t = let a=time() in

while time() -. a < t do prise\_event() done;;

(\*\*\*\*\*\*\*Fonction: revoie du signe de n\*\*\*\*\*\*\*)

let sgn n = if n<0 then -1 else 1;;

(\*\*\*\*\*\*\*Fonction: gestion des collisions\*\*\*\*\*\*\*)

let sion = function

|1->begin pax\_balle:=(sgn !pax\_balle)\* -1;

pay\_balle:=(sgn !pay\_balle)\* 1;

sound 700 20;

end

|2->begin pax\_balle:=(sgn !pax\_balle)\* -2;

pay\_balle:=(sgn !pay\_balle)\* 1;

sound 900 20;

end

|3->begin pax\_balle:=(sgn !pax\_balle)\* 1;

pay\_balle:=-1;

sound 700 20;

end

|4->begin pax\_balle:=(sgn !pax\_balle)\* 1;

pay\_balle:=(sgn !pay\_balle)\* -2;

sound 900 20;

end

|5->begin pax\_balle:= (sgn !pax\_balle)\*2;

pay\_balle:= 2;

sound 1000 20;

end

|6->begin pax\_balle:=(sgn !pax\_balle)\* 1;

pay\_balle:=1;

sound 500 20;

end

|7->begin pax\_balle:=-3;

pay\_balle:=1;

sound 500 20;

end

|8->begin pax\_balle:=3;

pay\_balle:=1;

sound 500 20;

end

|\_->();;

let coli ()=

(\* Sortie \*)

if !y\_balle < y\_barre then

begin decr vies;

x\_balle:= !x\_barre+ !lo\_barre/2;

y\_balle:=y\_barre+ha\_barre+ !r\_balle;

pay\_balle:=1;

sound 90 500;

vie(); end else

(\* Mur gauche-droite \*)

if !x\_balle+ !r\_balle+ !pax\_balle >= x\_ecran+lo\_ecran

|| !x\_balle- !r\_balle+ !pax\_balle <= x\_ecran

then sion 1 else

(\* Mur haut \*)

if !y\_balle+ !r\_balle+ 2\* !pay\_balle>=y\_ecran+ha\_ecran

then sion 3 else

(\* Barre milieu \*)

if !x\_balle >= !x\_barre+(4\* !lo\_barre)/10

&& !x\_balle <= !x\_barre+(6\* !lo\_barre)/10

&& !y\_balle- !r\_balle+ !pay\_balle <= y\_barre+ha\_barre

then sion 5 else

(\* Barre \*)

if((!x\_balle >= !x\_barre+(1\* !lo\_barre)/10

&& !x\_balle <= !x\_barre+(4\* !lo\_barre)/10)

|| (!x\_balle >= !x\_barre+(6\* !lo\_barre)/10

&& !x\_balle <= !x\_barre+(9\* !lo\_barre)/10))

&& !y\_balle- !r\_balle+ !pay\_balle <= y\_barre+ha\_barre

then sion 6 else

(\* Barre côté gauche \*)

if(!x\_balle >= !x\_barre

&& !x\_balle <= !x\_barre+(1\* !lo\_barre)/10

&& !y\_balle- !r\_balle- !pay\_balle <= y\_barre+ha\_barre)

||(!x\_balle+ !r\_balle+ !pax\_balle >= !x\_barre

&& !x\_balle- !r\_balle+ !pax\_balle <= !x\_barre

&& !y\_balle- !r\_balle+ !pay\_balle <= y\_barre+ha\_barre)

then sion 7 else

(\* Barre côté droit \*)

if(!x\_balle >= !x\_barre+(9\* !lo\_barre)/10

&& !x\_balle <= !x\_barre+ !lo\_barre

&& !y\_balle- !r\_balle+ !pay\_balle <= y\_barre+ha\_barre)

||(!x\_balle- !r\_balle+ !pax\_balle <= !x\_barre+ !lo\_barre

&& !x\_balle+ !r\_balle+ !pax\_balle >= !x\_barre+ !lo\_barre

&& !y\_balle- !r\_balle+ !pay\_balle <= y\_barre+ha\_barre)

then sion 8;

for k=0 to (nb\_bloc-1) do

(\* Bloc gauche-droite \*)

if(!x\_balle+ !r\_balle+ !pax\_balle = (extract bloc.(k) 1)

|| !x\_balle- !r\_balle+ !pax\_balle = (extract bloc.(k) 1)+lo\_bloc)

&& !y\_balle+ !r\_balle+ !pay\_balle >= (extract bloc.(k) 2)

&& !y\_balle- !r\_balle+ !pay\_balle <= (extract bloc.(k) 2)+ha\_bloc

then begin sion 2; efface k; end else

(\* Bloc haut-bas \*)

if(!y\_balle+ !r\_balle+ !pay\_balle = (extract bloc.(k) 2)

|| !y\_balle- !r\_balle+ !pay\_balle = (extract bloc.(k) 2)+ha\_bloc)

&& !x\_balle+ !r\_balle+ !pax\_balle >= (extract bloc.(k) 1)

&& !x\_balle- !r\_balle+ !pax\_balle <= (extract bloc.(k) 1)+lo\_bloc

then begin sion 4; efface k; end

done;;

(\*\*\*\*\*\*\*Fonction: déplacement de la balle\*\*\*\*\*\*\*)

let move\_balle () =

draw\_balle white;

coli();

x\_balle:=!x\_balle+ !pax\_balle;

y\_balle:=!y\_balle+ !pay\_balle;

draw\_balle blue;;

(\*\*\*\*\*\*\*Fonction: jeu\*\*\*\*\*\*\*)

let jeu () =

while !continuer do

wait 0.006;

draw\_barre();

move\_balle();

if ((int\_of\_float (time()\*. 100.)) mod 3000)=0 then hasard();

continuer:=false;

for k=0 to (nb\_bloc-1) do

if bloc.(k)<>(0,0) && !vies>0 then continuer:=true;

done;

done;;

(\*\*\*\*\*\*\*Fonction: lançement du jeu\*\*\*\*\*\*\*)

let jouer () =

open\_graph st\_ecran;

init();

jeu();

moveto (x\_ecran+lo\_ecran/2-50) (y\_ecran+ha\_ecran/2);

set\_color red;

if !vies=0 then begin draw\_string "Vous avez PERDU !!!"; sound 75 500; end

else if !vies<> -1 then begin draw\_string "Vous avez GAGNER !!!"; sound 600 500; end;

if !vies<> -1 then wait 2.;

close\_graph();;

jouer();;