```
(*******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();;
```