

```

dÃ©c. 19, 16 14:26 missionpacman.adb Page 1/2

with Ecran , Lab , Carte ;
use Ecran , Lab , Carte ;

procedure missionpacman is

  DureeJeu : constant Integer := 600;

  procedure AfficherGameOver Is
  begin
    EffacerEcran;
    EcrireEcran(12,7,"GAME OVER");
    EcrireEcran(17,10,"Appuyez sur la touche A");
    AttendreToucheA;
    EffacerEcran;
  end AfficherGameOver ;

  procedure AfficherVictoire (S : in Integer ) Is
  begin
    EffacerEcran;
    EcrireEcran(12,6,"Victoire !");
    EcrireEcran(12,8,"Score : " & Integer'Image(S*4));
    EcrireEcran(17,10,"Appuyez sur la touche A");
    AttendreToucheA;
    EffacerEcran;
  end AfficherVictoire ;

  procedure AfficherNombreVies (p : in T_Pacman ; L : in T_Lab) is
  begin
    SuspendreTimer ;
    EffacerEcran;
    EcrireEcran(12,2,"Plus que");
    EcrireEcran(17,5,Integer'Image(P.Nbrevies)&" vic(s)");
    EcrireEcran(17,10,"Appuyez sur la touche A");
    AttendreToucheA;
    EffacerEcran;
    DessinerLabyrinthe(L);
    ReprendreTimer ;
  end AfficherNombreVies ;

  function CompterCerise (L : in T_Lab) return Integer is
    Resultat : Integer := 0 ;
  begin
    for I in L'Range(1) loop
      for J in L'Range(2) loop
        if L(I,J) = Cerise then
          Resultat := Resultat +1 ;
        end if;
      end loop ;
    end loop ;
    return Resultat ;
  end CompterCerise ;

  procedure GererFinPartie (P : out T_Pacman ; L : out T_Lab) is
  begin
    InitialiserJeu(p,L);
    Mettreazerotimer ;
  end GererFinPartie ;

  procedure GererContactMur (P : in out T_Pacman ; L : in out T_Lab ) is
  begin
    P.NbreVies := P.NbreVies -1 ;
    if P.NbreVies > 0 then
      AfficherNombreVies(P,L);
    end if ;
  end GererContactMur ;

  procedure GererDeplacement( P : in out T_Pacman ; L : in out T_Lab ; D : in T_
Direction ; NbrC : in out Integer ) is
  begin
    DessinerBloc(P.PosX,P.PosY,Vide);
    L(P.PosX,P.PosY):=Vide;
    case D is
      when Sud =>
        P := (P.PosX,P.PosY+1,P.NbreVies);
      when Nord =>
        P := (P.PosX,P.PosY-1,P.NbreVies);
      when Est =>
        P := (P.PosX+1,P.PosY,P.NbreVies);
      when Ouest =>
        P := (P.PosX-1,P.PosY,P.NbreVies);
      when others =>
        null;
      end case ;
    if L(P.PosX,P.PosY) = Cerise then
      NbrC := NbrC +1 ;
    end if;
    L(P.PosX,P.PosY):=Pacman;
    DessinerBloc(P.PosX,P.PosY,Pacman, D);
  end GererDeplacement ;

  procedure AfficherTemps is
  begin
    EcrireEcran(32,4,"Decompte");
    -- if Float(DureeJeu-GetTempsEcoule) < 0.2 * Float(DureeJeu) then
    --   Insa.Graphics.SetTextColor(Red);
    -- end if;
    EcrireEcran(35,6,Integer'Image(DureeJeu-GetTempsEcoule) & " ");
  end AfficherTemps;

  function TesterMur (D: T_Direction ; L : T_Lab ; P : T_Pacman ) return Boolean
  is
    CestMur : Boolean := False ;
  begin
    if (D = Sud and L(P.PosX,P.PosY+1) = Mur) or
       (D = Nord and L(P.PosX,P.PosY-1) = Mur) or
       (D = Ouest and L(P.PosX-1,P.PosY) = Mur) or
       (D = Est and L(P.PosX+1,P.PosY) = Mur) then
      CestMur := True ;
    end if;
    return CestMur ;
  end TesterMur ;

  Mon_Lab : T_Lab ;
  Mon_Pacman : T_Pacman ;
  Ma_Direction : T_Direction ;
  Mon_Nbre_Cerises_Mangees,Mon_Nbre_Cerises_Depart : Integer := 0 ;

begin
  Lab.InitialiserJeu(Mon_Pacman,Mon_Lab);
  Carte.InitialiserCarte ;
  Mon_Nbre_Cerises_Depart := CompterCerise(Mon_Lab);
  while True loop
    while (getTempsEcoule <= DureeJeu) and (Mon_Nbre_Cerises_Mangees < Mon_Nbr
e_Cerises_Depart) and (Mon_Pacman.NbreVies > 0) loop
      AfficherTemps ;
      Ma_Direction := Carte.DetecterDirection ;
      if TesterMur(Ma_Direction,Mon_Lab,Mon_Pacman) then
        GererContactMur(Mon_Pacman,Mon_Lab);
      elsif Ma_Direction /= Immobile then
        GererDeplacement(Mon_Pacman,Mon_Lab,Ma_Direction,Mon_Nbre_Cerises_Ma
ngees);
      end if;
    end loop;
    if (getTempsEcoule > DureeJeu) or (Mon_Pacman.NbreVies <= 0) then
      AfficherGameOver ;

```

```

dÃ©c. 19, 16 14:26 missionpacman.adb Page 2/2

    else
      AfficherVictoire(DureeJeu - getTempsEcoule) ;
    end if;
    GererFinPartie(Mon_Pacman,Mon_Lab);
    Mon_Nbre_Cerises_Depart := CompterCerise(Mon_Lab);
    Mon_Nbre_Cerises_Mangees := 0 ;
  end loop ;

end Missionpacman ;

```

```
pragma Ada_95;

package carte is

  type T_Direction is (Nord,Sud,Est,Ouest,Immobile) ;

  -- fonction devant etre appelee imperativement au debut de votre programme
  procedure InitialiserCarte ;

  -- retourne un entier qui represente le nombre de 100ms
  -- ecoules depuis l'appel de la procedure InitialiserCarte
  function GetTempsEcoule return Integer ;

  -- fige le temps !!!
  procedure SuspendreTimer ;

  -- "defige" le temps !!!
  procedure ReprendreTimer ;

  -- redemarre le temps Ã partir de '0'
  procedure MettreAZeroTimer ;

  -- attend un appui sur la touche A
  procedure AttendreToucheA ;

  -- renvoie la derniere touche de direction appuyee de type T_Direction
  function DetecterDirection return T_Direction ;

end carte ;
```

```
with Insa , Insa.Graphics ;
use Insa , Insa.Graphics ;

package Ecran is

  -- Efface un ecran et fixe un fond bleu fonce
  procedure EffacerEcran ;

  -- Ecrit la chaine S en blan sur fond bleu fonce
  -- avec le 1er caractere a la colonne C et ligne L
  -- C appartient a [0..39] et L appartient a [0..14]
  procedure EcrireEcran (C : in Integer;L : in Integer;S : in String) ;

end Ecran;
```

```

pragma Ada_95;

with Carte;
use Carte;

package Lab is

    MaxNbVies : constant Integer := 4 ;

    type T_Bloc is (Mur,Vide,Pacman,Cerise) ;

    type T_Lab is array (1..10,1..10) of T_Bloc ;

    type T_Pacman is record
        PosX : Integer ;
        PosY : Integer ;
        NbVies : Integer ;
    end record ;

    Invalid_Bloc: exception;

    -- Efface l'ecran, genere et dessine un labyrinthe
    -- Positionne Pacman avec MaxNbVies vies au milieu du labyrinthe
    procedure InitialiserJeu (P : out T_Pacman ; L : out T_Lab) ;

    -- Retourne une variable de type T_Lab initialisee avec un labyrinthe
    -- dans lequel un T_Pacman P est place
    function GenererLabyrinthe(P : T_Pacman) return T_Lab ;

    -- Dessine a la Ieme ligne Jeme colonne du labyrinthe un T_Bloc defini par Ty
    peBloc
    procedure DessinerBloc (I,J : Integer ; TypeBloc : T_Bloc) ;

    -- Dessine a la Ieme ligne Jeme colonne du labyrinthe un bloc pacman (Pacman_
    Haut, Pacman_Droit, Pacman_Gauche, Pacman_Bas)
    -- en fonction de la direction.
    -- Si le bloc n'est pas un bloc pacman, une exception Invalid_Bloc est levÃ©e
    procedure DessinerBloc (I,J : Integer ; TypeBloc : T_Bloc; Direction: T_Directi
    on );

    -- Dessine un labyrinthe T_Lab defini par L
    procedure DessinerLabyrinthe (L : T_Lab) ;

end Lab;

```