

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

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

  -- renvoie la derniere direction observee par les accelerometres
  function DetecterDirection return T_Direction ;

  -- 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) ;

  TempsEcoule : integer ;
end carte ;
```

```

package plateau is

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

  type T_Table is array (0..11,0..15) of T_Bloc ;

  type T_Snake is record
    PosX : Integer ;
    PosY : Integer ;
  end record ;

  type Element ;
  type P_Element is access Element ;
  type Element is record
    Serpent : T_Snake ;
    Suiv : P_Element ;
  end record;

  -- Retourne T de type T_Table avec toutes les case vides
  -- et la position du serpent S au milieu
  -- affiche ce serpent
  procedure InitialiserJeu (T: out T_Table ; S: out T_Snake) ;

  -- Dessine a la Ieme ligne Jeme colonne de la table un T_Bloc defini par Type
  Bloc
  procedure DessinerBloc (I,J : Integer ; TypeBloc : T_Bloc) ;

  -- Libere l'espace memoire reserve a un pointeur L
  -- Attention il est indispensable d'appeler cette procedure
  -- Pour faire une desallocation "propre" de pointeur
  -- pour ne pas saturer la memoire de la carte
  procedure EffacerMemoireElement (L : in out P_Element) ;

  -- placer une cerise aleatoirement
  -- dans la table T et a l'ecran
  procedure PlacerCerise (T:in out T_Table);

end plateau;

```