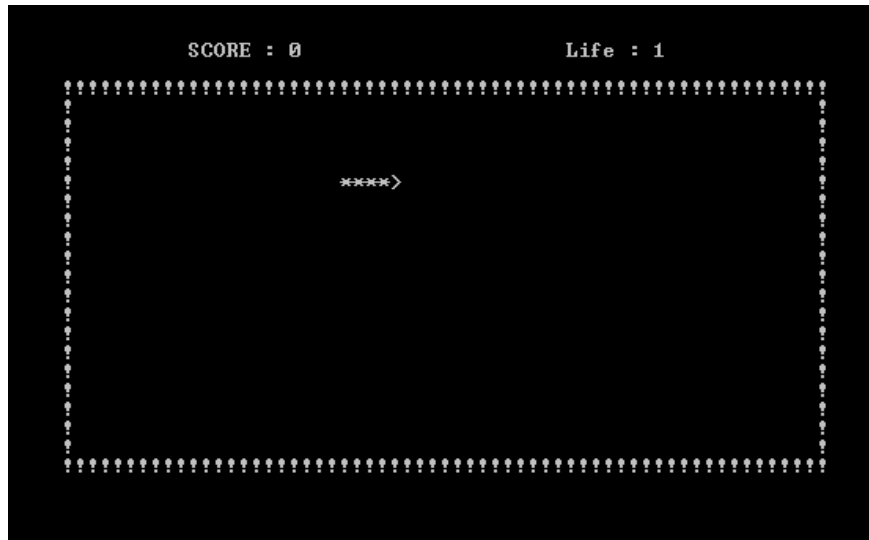


CPI1 2015/2016

Mini-projet

Serpent



« Le serpent » est un jeu consistant à piloter un serpent virtuel qui grandit, à lui faire avaler des bonus, et à éviter que sa tête ne rentre en collision avec l'environnement ou lui-même. Les listes chaînées sont idéales pour modéliser un tel serpent : on stocke dans une liste les cases occupées par le serpent, et à chaque itération la nouvelle position de la tête est ajoutée en tête de liste alors que la position de la queue est supprimée en fin de liste (sauf si le serpent est en train de grandir).

Nous définissons donc les types suivants :

```
type noeud = ^noeud;
noeud = record
    x, y: integer; // les coordonnées d'une case occupée
    suivant: noeud;
end;

type serpent = record
    taille: integer; // la taille actuelle du serpent
    direction: integer; // 0=up, 1=right, 2=down, 3=left
    tete: noeud; // la tete du serpent
end;
```

Partie obligatoire

Ecrire un programme qui permet à l'utilisateur de piloter un serpent à l'aide des quatre flèches du clavier. Tous les n pas de temps, le serpent grandira.

L'exemple fourni à la fin de l'énoncé vous montre comment détecter les touches appuyées par l'utilisateur dans votre boucle de jeu. A chaque pas de temps, vous pouvez utiliser les deux fonctions suivantes de la unit crt :

- delay (en ms) pour ralentir éventuellement votre simulation
- clrscr pour effacer la console et donner l'illusion d'un serpent animé.

Concernant l'architecture logicielle, vous devrez implémenter

- une unit Serpent avec toutes les fonctionnalités nécessaires (initialiser, déplacer, grandir, libérer, etc.)
- un programme principal qui gère la boucle du jeu et manipule le serpent
- un programme de tests unitaires pour la unit Serpent.

Partie libre

- 1) Ajouter une collision quand le serpent touche les bords du terrain.
- 2) Ajouter une collision quand le serpent rentre en contact avec lui-même.
- 3) Ajouter des bonus et des malus, des vies, un score etc.

Planning

Date de rendu: dimanche 17 avril 2016.

Au plus tard ce jour-là à minuit, vous devez avoir déposé sur AREL une archive zip/tar contenant les sources Pascal de votre projet ainsi qu'un petit rapport (format pdf).

```
program ExampleKeyPressed;
uses crt;
// Ce programme affiche le code ascii
// des touches appuyées par l'utilisateur.
// Notez que les flèches font partie du code étendu
// (#0 puis un deuxième code ascii)
var c : char;
begin
  c:=#0;
  while c<>#27 do
    begin
      if keypressed then begin
        c:=readkey;
        if c=#0 then // if extended code
          begin
            c:=readkey; // read the code once more
            writeln('Extended : ', ord(c));
          end
        else writeln(ord(c));
      end else write('.');
    end;
end.
```