9/7.5

Le serpent mange-pierres

Un serpent se déplace à vitesse constante sur l'écran. Il doit avaler toutes les pierres et éviter les obstacles tels que les bords de l'écran et la barre qui se trouve au centre de l'écran. Tout contact avec une zone interdite met fin au programme.

Le programme est proposé en Basic et en Turbo Pascal.

Comment fonctionne le programme

Si vous désirez utiliser la version Basic du programme, saisissez le listing suivant :

```
1000 REM ========================
1010 REM Le serpent mange-pierre
1020 REM ========================
1030 REM
1040 GOSUB 2000 ' Initialisation de la partie
1050 GOSUB 3000 ' Jeu
1060 GOSUB 4000 ' Commentaire
1070 END
1080 '
2000 REM ---------------------------
2010 REM Initialisation d'une partie
2020 REM ---------------------------
2030 REM
2040 CLS
2050 FOR i=1 TO 10
2060   x(i)=INT(RND(1)*78)+2
2070   y(i)=INT(RND(1)*23)+2 : IF y(i)=12 THEN 2070
2080   FOR j=1 TO i-1
2090     IF (x(i)=x(j)) AND (y(i)=y(j)) THEN i=i-1
2100   NEXT j
2110 NEXT i
```

```
2120 LOCATE 20,12
2130 PRINT STRING$(40,CHR$(207))
2140 FOR i=1 TO 10
2150   LOCATE x(i),y(i)
2160   PRINT CHR$(238)
2170 NEXT i
2180 LOCATE 1,1
2190 PRINT CHR$(150)+STRING$(78,CHR$(154))+CHR$(156)
2200 FOR i=2 TO 24
2210   LOCATE 1,i
2220   PRINT CHR$(149)
2230   LOCATE 80,i
2240   PRINT CHR$(149)
2250 NEXT i
2260 LOCATE 1,25
2270 PRINT CHR$(147)+STRING$(78,CHR$(154))+CHR$(153);
2280 LOCATE 1,1
2290 RETURN
2300 REM
3000 REM ------------------------
3010 REM Deroulement d'une partie
3020 REM ------------------------
3030 REM
3040 c1$=CHR$(231) 'Corps
3050 c2$=CHR$(224) 'Tete souriante
3060 s$(1)=c1$:s$(2)=c1$:s$(3)=c1$:s$(4)=c1$:s$(5)=c1$:s$(6)
=c2$ ' Le serpent
3070 LOCATE 2,2
3080 PRINT STRING$(5,c1$)+c2$
3090 dir=1 'Vers la droite
3100 FOR i=1 TO 6
3110   READ xs(i),ys(i)
3120 NEXT i
3130 DATA 2,2,3,2,4,2,5,2,6,2,7,2
3140 a$=INKEY$
3150 GOSUB 3240 'Deplacement
3160 IF a$<>"" THEN GOSUB 3560 'Changement de direction
3170 IF fin=0 THEN 3140
3180 RETURN
3190 '
3200 ' ------------------------------
3210 ' Deplacement en fonction de dir
3220 ' ------------------------------
3230 '
3240 j=0
3250 LOCATE xs(1),ys(1):PRINT" "
3260 FOR i=1 TO 5
3270   ys(i)=ys(i+1)
```

```
3280   xs(i)=xs(i+1)
3290 NEXT i
3300 IF dir=1 THEN xs(6)=xs(6)+1 'Vers la droite
3310 IF dir=2 THEN ys(6)=ys(6)+1 'Vers le bas
3320 IF dir=3 THEN xs(6)=xs(6)-1 'Vers la gauche
3330 IF dir=4 THEN ys(6)=ys(6)-1 'Vers le haut
3340 IF j<>0 THEN ys(j)=ys(6)
3350 '
3360 ' ------------------
3370 ' Test de collision
3380 ' ------------------
3390 '
3400 FOR i=1 TO 6
3410   IF xs(i)=1 OR xs(i)=80 THEN fin=1
3420   IF ys(i)=1 OR ys(i)=25 THEN fin=1
3430   IF ys(i)=12 AND xs(i)>19 AND xs(i)<41 THEN fin=1
3440 NEXT i
3450 IF fin THEN RETURN
3460 FOR i=1 TO 6
3470   LOCATE xs(i),ys(i)
3480   PRINT s$(i)
3490 NEXT i
3500 RETURN
3510 '
3520 ' ------------------------
3530 ' Changement de direction
3540 ' ------------------------
3550 '
3560 a=ASC(a$)
3570 IF a=240 THEN dir=4 'Vers le haut
3580 IF a=243 THEN dir=1 'Vers la droite
3590 IF a=241 THEN dir=2 'Vers le bas
3600 IF a=242 THEN dir=3 'Vers la gauche
3610 RETURN
3620 REM
3630 REM Test de collision
3640 REM
4000 REM -----------
4010 REM Fin du jeu
4020 REM -----------
4030 REM
4040 INK 0,2,12
4050 FOR i=1 TO 2000
4060 NEXT i
4070 INK 0,0:BORDER 0
4080 CLS
4090 PRINT "Vous etes mort par indigestion."
4100 RETURN
```

Si vous désirez utiliser la version Turbo Pascal du programme, saisissez le listing suivant :

```
Program Serpent;
{------------------------ }
{ Le serpent mange-pierre }
{------------------------ }
Var
  I,J : Integer;                 { Index de boucles                    }
  X,Y : Array[1..10] of byte;    { Position des rochers sur l'ecran    }
  dir : Byte;                    { Direction du serpent                }
  xs,ys : Array[1..6] of Byte;   { Position du serpent                 }
  Fin : Byte;                    { Indicateur de fin de partie         }
  S : Array[1..6] of Char;       { Le corps du serpent                 }
  c1, c2 : Char;                 { Caracteres intermediaires           }
  a : char;                      { Caractere tape au clavier           }
  KP : Boolean;                  { Indicateur de touche pressee        }


Procedure Initialisation;
{ --------------------------- }
{ Initialisation d'une partie  }
{ --------------------------- }
begin
  ClrScr;
  FOR i:=1 TO 10 do
  begin
    x[i]:=Round(Random*78)+2;
    Repeat
      y[i]:=Round(Random*22)+2;
    until y[i]<>12;
    FOR j:=1 TO i-1 do
      IF (x[i]=x[j]) AND (y[i]=y[j]) THEN i:=i-1;
  end;
  GotoXY(20,12);
  For I:=1 to 40 do
    Write(CHR(207));
  FOR i:=1 TO 10 do
  begin
    GotoXY(x[i],y[i]);
    Write(CHR(238));
  end;
  GotoXY(1,1);
  Write(CHR(150));
  For I:=1 to 78 do
    Write(CHR(154));
  Write(CHR(156));
  FOR i:=2 TO 23 do
  begin
```

```
  GotoXY(1,i);
  Write(CHR(149));
  GotoXY(80,i);
  Write(CHR(149));
  end;
  GotoXY(1,24);
  Write(CHR(147));
  For i:=1 to 78 do
    Write(CHR(154));
  Write(CHR(153));
  GotoXY(1,1);
  Fin:=0;
end;


Procedure Deplacement;
{ ------------------------------ }
{ Deplacement en fonction de dir  }
{ ------------------------------ }
begin
  j:=0;
  GotoXY(xs[1],ys[1]);
  Write(' ');
  FOR i:=1 TO 5 do
  begin
    ys[i]:=ys[i+1];
    xs[i]:=xs[i+1];
  end;
  IF dir=1 THEN
     xs[6]:=xs[6]+1; {Vers la droite }
  IF dir=2 THEN
     ys[6]:=ys[6]+1; {Vers le bas }
  IF dir=3 THEN
     xs[6]:=xs[6]-1; {Vers la gauche }
  IF dir=4 THEN
     ys[6]:=ys[6]-1; {Vers le haut }
  IF j<>0 THEN ys[j]:=ys[6];

  { ------------------ }
  { Test de collision  }
  { ------------------ }
  FOR i:=1 TO 6 do
  begin
    IF (xs[i]=1) OR (xs[i]=80) THEN
       fin:=1;
```

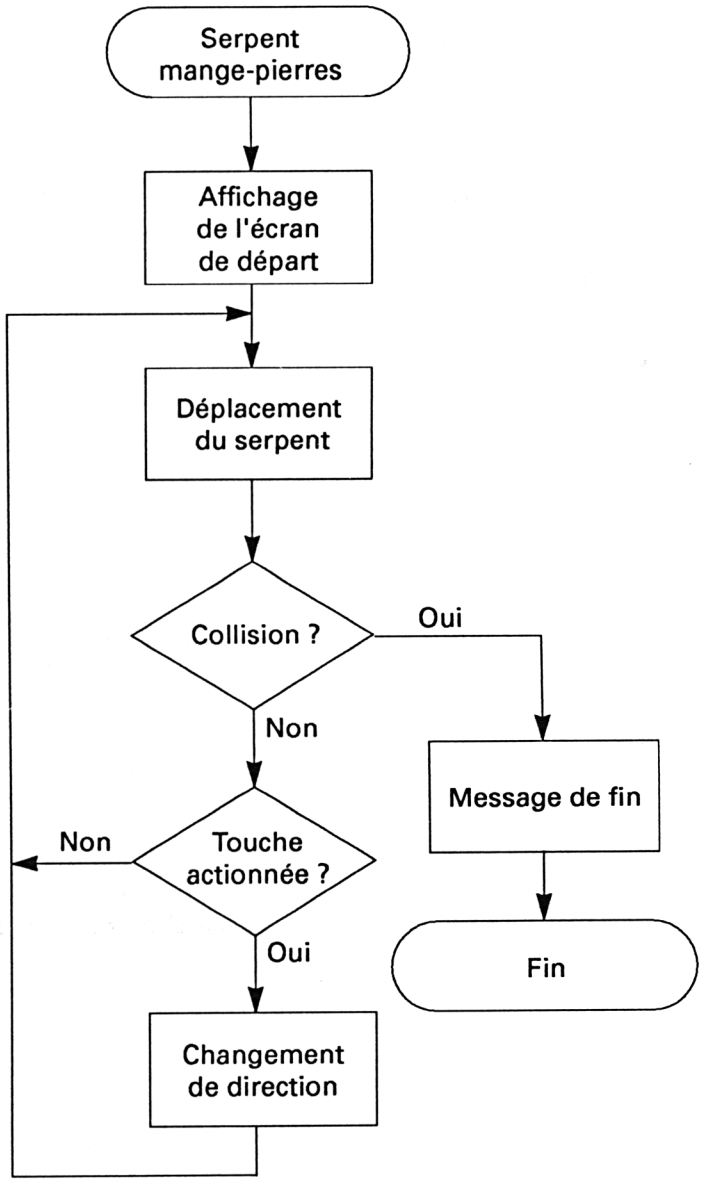
```
    IF (ys[i]=1) OR (ys[i]=25) THEN
       fin:=1;
    IF (ys[i]=12) AND (xs[i]>19) AND (xs[i]<61) THEN
       fin:=1;
  end;
  IF fin=0 THEN
  FOR i:=1 TO 6 do
  begin
    GotoXY(xs[i],ys[i]);
    Write(s[i]);
  end;
end;


Procedure Direction;
{ ------------------------ }
{ Changement de direction  }
{ ------------------------ }
begin
  IF ord(a)=240 THEN
     dir:=4; {Vers le haut }
  IF ord(a)=243 THEN
     dir:=1; {Vers la droite }
  IF ord(a)=241 THEN
     dir:=2; {Vers le bas }
  IF ord(a)=242 THEN
     dir:=3; {Vers la gauche }
end;


Procedure Jeu;
{ ------------------------- }
{ Deroulement d'une partie  }
{ ------------------------- }
begin
  c1:=CHR(231); { Corps           }
  c2:=CHR(224); { Tete souriante }
  s[1]:=c1;
  s[2]:=c1;
  s[3]:=c1;
  s[4]:=c1;
  s[5]:=c1;
  s[6]:=c2;  { Le serpent     }
  dir:=1;    { Vers la droite }
  xs[1]:=2; ys[1]:=2;
  xs[2]:=3; ys[2]:=2;
```

```
    xs[3]:=4; ys[3]:=2;
    xs[4]:=5; ys[4]:=2;
    xs[5]:=6; ys[5]:=2;
    xs[6]:=7; ys[6]:=2;
    Repeat
      KP:=False;
      Deplacement;  { Deplacement du serpent }
      For I:=1 to 30 do
        If KeyPressed then KP:=True;
      If KP then
      begin
      Read(Kbd,a);
      Direction; { Changement de direction }
    end;
  until Fin<>0;
end;


Procedure Commentaire;
{ ----------- }
{ Fin du jeu  }
{ ----------- }
begin
  ClrScr;
  Writeln('Vous etes mort par indigestion.');
end;

                        {---------------------}
                        { PROGRAMME PRINCIPAL }
                        {---------------------}
begin
  Initialisation;  { Initialisation de la partie  }
  Jeu;             { Declenchement du jeu         }
  Commentaire;     { Commentaire de fin de partie }
end.
```

Lancez le programme. Dix rochers sont affichés à des positions aléatoires sur l'écran. Utilisez les touches flèches du clavier pour modifier la direction du serpent. Evitez scrupuleusement l'obstacle central et les bords de l'écran, sinon, c'en est fini de vous...

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme ci-dessous :



Analysons la version Turbo Pascal.

Le programme débute par la section des variables.

Signalons en particulier :

– les tableaux **X** et **Y** qui contiennent la position des rochers sur l'écran ;
– les tableaux **XS** et **YS** qui contiennent la position des six anneaux du serpent ;
– enfin, le tableau de caractères **S** qui contient les anneaux du serpent.

Le programme se poursuit par la section des procédures.

La procédure **Initialisation** est responsable :

– du tirage aléatoire de la position des rochers ;
– de l'affichage du jeu au départ.

Les rochers sont positionnés de telle sorte qu'ils ne se trouvent pas sur la 12e ligne de l'écran. Cette ligne contient en effet un obstacle infranchissable :

```
Repeat
  y [i] :=Round (Random*22)+2 ;
Until y [i] <>12 ;
```

Les rochers sont représentés à l'aide du caractère de code ASCII **238** :

```
For i:=1 to 10 do
begin
  GotoXY (x [i], y [i]) ;
  Write (CHR (238)) ;
end ;
```

Enfin, le cadre du jeu est tracé à l'aide de caractères semi-graphiques de codes ASCII **149, 147, 154, 153, 150** et **156**.

Le programme se poursuit par la procédure **Deplacement**. Cette procédure permet de déplacer le serpent sur l'écran. En fonction de sa direction (variable **dir**), elle :

– calcule la position des six anneaux dans les tableaux **XS** et **YS** ;
– efface l'ancienne position du serpent ;
– affiche le serpent dans sa nouvelle position.

Cette procédure teste également la collision entre un des anneaux du serpent et la barre centrale ou un des bords de l'écran. Si un des anneaux touche une zone interdite, la variable fin est initialisée à 1, ce qui provoque la fin du programme.

La procédure **Direction** est activée lorsqu'une touche du clavier est pressée. Si la touche pressée est une touche flèche, elle modifie la direction du serpent en agissant sur la variable **dir** :

```
IF ord (a)= 240 THEN
  dir:=4 ; {Vers le haut}
...
IF ord (a)=242 THEN
  dir:=3 ; {Vers la gauche}
```

Le programme se poursuit par la procédure principale : la procédure **Jeu**. Ses fonctions sont doubles :

– activation périodique de la procédure **Deplacement** ;
– activation de la procédure **Direction** lorsqu'une touche du clavier est frappée.

Les premières lignes de la procédure définissent la forme et la position du serpent au départ. Les lignes suivantes contiennent la boucle principale du jeu. De type **Repeat Until**, cette boucle prend fin lorsque le serpent entre en contact avec une zone interdite. Dans ce cas, la variable **Fin** est égale à 1 :

```
Repeat
  ...
Until Fin<>0 ;
```

L'acquisition des touches frappées au clavier se fait dans une boucle **For** :

```
For I:=1 to 30 do
  If KeyPressed then KP:=True ;
```

Augmentez la borne supérieure de la boucle pour rendre le serpent plus sensible, et diminuez-là pour obtenir l'effet inverse.

La dernière procédure affiche le message

Vous êtes mort par indigestion

lorsque le serpent entre en contact avec une zone interdite. Enfin, le programme se termine par la section du programme principal qui active séquentiellement les trois procédures principales du programme :

```
Initialisation ;    {Initialisation de la partie      }
Jeu ;               { Déclenchement du jeu            }
Commentaire ;       { Commentaire de fin de partie    }
```