

## 9/9.5

# Jeu de la vie

---

La multiplication ou la disparition des cellules d'un être vivant est liée à des lois bien précises. A l'aide de ce programme, nous allons visualiser l'évolution d'un groupe de cellules animales dont les lois de survie sont les suivantes :

- toute cellule entourée par deux ou trois autres cellules survit (adaptation) ;
- toute cellule entourée par trois autres cellules se reproduit et donne naissance à une nouvelle cellule (reproduction) ;
- toute cellule entourée par une, quatre cellules ou plus meurt (sous-ou surpopulation).

**COMMENT UTILISER LE PROGRAMME ?**

Le programme a été développé en Basic et en Turbo Pascal. En voici les listings. Listing de la version Basic :

```

1000 REM =====
1010 REM Jeu de la vie
1020 REM =====
1030 REM
1040 GOSUB 1090 'Initialisation
1050 GOSUB 2180 'Calcul des generations
1060 GOSUB 2780 'Fin du jeu
1070 END
1080 REM
1090 REM -----
1100 REM Initialisation du jeu
1110 REM -----
1120 REM
1130 DIM t(10,10)
1140 MODE 1
1150 PRINT"          Jeu de la vie"
1160 PRINT"          -----"
1170 PRINT:PRINT
1180 PRINT"Vous allez voir naitre et mourir"
1190 PRINT"sous vos yeux une population de"
1200 PRINT"cellules."
1210 PRINT
1220 PRINT"Son evolution est dictée par les"
1230 PRINT"lois suivantes :"
1240 PRINT
1250 PRINT"- toute cellule entouree par deux"
1260 PRINT" ou trois autres cellules survit,"
1270 PRINT
1280 PRINT"- lorsqu'une cellule est entouree"
1290 PRINT" de trois autres, une nouvelle"
1300 PRINT" cellule nait,"
1310 PRINT
1320 PRINT"- toute cellule entouree par une,"
1330 PRINT" quatre cellules ou plus meurt."
1340 PRINT:PRINT
1350 PRINT" Appuyez sur une touche ..."
1360 ch$=INKEY$
1370 IF ch$="" THEN 1360
1380 CLS
1390 FOR i=1 TO 10
1400 LOCATE 10,i+5
1410   FOR j=1 TO 10
1420     PRINT". ";
1430   NEXT j
1440 NEXT i
1450 LOCATE 1,19
1460 PRINT"Deplacez le curseur a l'aide des"
1470 PRINT"touches fleches. Appuyez sur <F1>"
1480 PRINT"pour placer une cellule et sur"
1490 PRINT"<F2> pour en enlever une. Appuyez"
1500 PRINT"sur <Enter> pour valider les"
1510 PRINT"donnees entrees."

```

```

1520 REM
1530 x=1:y=1:LOCATE 10,y+5:PRINT CHR$(159)
1540 REM Boucle de saisie
1550   ch$=INKEY$
1560   IF ch$="" THEN 1550
1570   IF ASC(ch$)<>243 THEN 1680
1580   REM
1590   REM - - - - -
1600   REM Deplacement vers la droite
1610   REM - - - - -
1620   REM
1630   IF x=10 THEN 1540
1640   c=t(x,y)
1650   LOCATE 8+x*2,y+5
1660   IF c=0 THEN PRINT"." ELSE PRINT CHR$(230)
1670   x=x+1:LOCATE 8+x*2,y+5:PRINT CHR$(159)
1680   IF ASC(ch$)<>242 THEN 1790
1690   REM
1700   REM - - - - -
1710   REM Deplacement vers la gauche
1720   REM - - - - -
1730   REM
1740   IF x=1 THEN 1540
1750   c=t(x,y)
1760   LOCATE 8+x*2,y+5
1770   IF c=0 THEN PRINT"." ELSE PRINT CHR$(230)
1780   x=x-1:LOCATE 8+x*2,y+5:PRINT CHR$(159)
1790   IF ASC(ch$)<>240 THEN 1900
1800   REM
1810   REM - - - - -
1820   REM Deplacement vers le haut
1830   REM - - - - -
1840   REM
1850   IF y=1 THEN 1540
1860   c=t(x,y)
1870   LOCATE 8+x*2,y+5
1880   IF c=0 THEN PRINT"." ELSE PRINT CHR$(230)
1890   y=y-1:LOCATE 8+x*2,y+5:PRINT CHR$(159)
1900   IF ASC(ch$)<>241 THEN 2010
1910   REM
1920   REM - - - - -
1930   REM Deplacement vers le bas
1940   REM - - - - -
1950   REM
1960   IF y=10 THEN 1540
1970   c=t(x,y)
1980   LOCATE 8+x*2,y+5
1990   IF c=0 THEN PRINT"." ELSE PRINT CHR$(230)
2000   y=y+1:LOCATE 8+x*2,y+5:PRINT CHR$(159)
2010   IF ASC(ch$)<>49 THEN 2080
2020   REM
2030   REM - - - - -
2040   REM Placement d'une cellule
2050   REM - - - - -

```

```

2060 REM
2070 t(x,y)=1
2080 IF ASC(ch#)<>50 THEN 2150
2090 REM
2100 REM - - - - -
2110 REM Effacement d'une cellule
2120 REM - - - - -
2130 REM
2140 t(x,y)=0
2150 IF ASC(ch#)<>13 THEN 1550
2160 RETURN
2170 REM
2180 REM -----
2190 REM Evolution de la comunaute
2200 REM -----
2210 REM
2220 CLS : GOSUB 2640 'Affichage de la iere generatiou
2230 GOSUB 2290 'Calcul de la prochaine generation
2240 GOSUB 2640 'Affichage de la prochaine generation
2250 ch#=INKEY#
2260 IF UPPER$(ch#)="F" THEN RETURN
2270 GOTO 2230
2280 REM
2290 REM - - - - -
2300 REM Calcul de la prochaine generation
2310 REM - - - - -
2320 REM
2330 gen=gen+1
2340 FOR i=2 TO 8
2350   FOR j=2 TO 8
2360     r=0
2370     IF t(i-1,j-1)=1 THEN r=r+1
2380     IF t(i-1,j)=1 THEN r=r+1
2390     IF t(i-1,j+1)=1 THEN r=r+1
2400     IF t(i,j-1)=1 THEN r=r+1
2410     IF t(i,j+1)=1 THEN r=r+1
2420     IF t(i+1,j-1)=1 THEN r=r+1
2430     IF t(i+1,j)=1 THEN r=r+1
2440     IF t(i+1,j+1)=1 THEN r=r+1
2450     IF t(i,j)=1 THEN GOSUB 2500 ELSE GOSUB 2570
2460   NEXT j
2470 NEXT i
2480 RETURN
2490 REM
2500 REM - - - - -
2510 REM Mort eventuelle d'une cellule
2520 REM - - - - -
2530 REM
2540 IF r<>2 AND r<>3 THEN t(i,j)=0
2550 RETURN
2560 REM
2570 REM - - - - -
2580 REM Naissance eventuelle d'une cellule
2590 REM - - - - -

```

```
2600 REM
2610 IF r=3 THEN t(i,j)=1
2620 RETURN
2630 REM
2640 REM - - - - -
2650 REM Affichage de la prochaine generation
2660 REM - - - - -
2670 REM
2680 LOCATE 11,1:PRINT"Generation ";gen
2690 LOCATE 11,2:PRINT"-----"
2700 FOR y=1 TO 10
2710   LOCATE 10,y+5
2720   FOR x=1 TO 10
2730     IF t(x,y)=1 THEN PRINT CHR$(230);" "; ELSE PRINT "
";
2740   NEXT x
2750 NEXT y
2760 RETURN
2770 REM
2780 REM -----
2790 REM Fin de la simulation
2800 REM -----
2810 REM
2820 TOT=0
2830 FOR i=1 TO 10
2840   FOR j=1 TO 10
2850     TOT=TOT+T(I,J)
2860   NEXT j
2870 NEXT i
2880 REM
2890 LOCATE 1,20
2900 IF TOT=0 THEN PRINT"Population decimee en";gen;"generat
ion(s). "
2910 IF TOT<>0 THEN PRINT"Fin de la simulation a la generati
on";gen
2920 RETURN
```

Listing de la version Turbo Pascal :

```
Program Jeu_de_la_Vie;

Var T : Array[1..10,1..10] of Byte;~
    C,X,Y,I,J,R,Gen,Tot : Byte;
    Ch : Char;

Procedure Initialisation;
begin
  ClrScr;
  Writeln('          Jeu de la vie');
  Writeln('          -----');
  Writeln;
  Writeln;
  Writeln('Vous allez voir naitre et mourir');
  Writeln('sous vos yeux une population de ');
  Writeln('cellules. ');
  Writeln;
  Writeln('Son evolution est dictee par les');
  Writeln('lois suivantes : ');
  Writeln;
  Writeln('- toute cellule entouree par deux');
  Writeln(' ou trois autres cellules survit, ');
  Writeln;
  Writeln('- lorsqu''une cellule est entouree');
  Writeln(' de trois autres, une nouvelle');
  Writeln(' cellule nait, ');
  Writeln;
  Writeln('- toute cellule entouree par une, ');
  Writeln(' quatre cellules ou plus meurt. ');
  Writeln;
  Writeln;
  Write(' Appuyez sur une touche ... ');
  Repeat
  Until KeyPressed;
  Read(Kbd,Ch);

  ClrScr;
  Gen:=0;
  For I:=1 to 10 do
    For J:=1 to 10 do
      T[I,J]:=0;

  For I:=1 to 10 do
  begin
    GotoXY(10,i+5);
    For J:=1 to 10 do
      Write('. ');
```

```

end;
GotoXY(1,19);
Writeln('Deplacez le curseur a l''aide des');
Writeln('touches fleches. Appuyez sur <F1>');
Writeln('pour placer une cellule, et sur');
Writeln('<F2> pour en enlever une. Appuyez');
Writeln('sur <Enter> pour valider les');
Writeln('donnees entrees');

X:=1; Y:=1;
GotoXY(10,y+5);
Write(Chr(159));

Repeat
  While not Keypressed do;
  Read(Kbd,Ch);

  Case Ord(Ch) of

    243 : begin
      if (X<>10) then
      begin
        c:=T[X,Y];
        GotoXY(8+X*2,Y+5);
        If (C=0) then Write('.')
          else Write(Chr(230));
        X:=X+1;
        GotoXY(8+X*2,Y+5);
        Write(Chr(159));
      end;
    end;

    242 : begin
      if (X<>1) then
      begin
        c:=T[X,Y];
        GotoXY(8+X*2,Y+5);
        If (C=0) then Write('.')
          else Write(Chr(230));
        X:=X-1;
        GotoXY(8+X*2,Y+5);
        Write(Chr(159));
      end;
    end;

    240 : begin
      if (Y<>1) then
      begin
        c:=T[X,Y];
        GotoXY(8+X*2,Y+5);
        If (C=0) then Write('.')
          else Write(Chr(230));
        Y:=Y-1;
      end;
    end;
  end;
end;

```

```

                GotoXY(8+X*2,Y+5);
                Write(Chr(159));
            end;
        end;
        241 : begin
                if (Y<>10) then
                begin
                    c:=T[X,Y];
                    GotoXY(8+X*2,Y+5);
                    If (C=0) then Write('.')
                    else Write(Chr(230));
                    Y:=Y+1;
                    GotoXY(8+X*2,Y+5);
                    Write(Chr(159));
                end;
            end;

            49  : T[X,Y]:=1;

            50  : T[X,Y]:=0;
        end;
    until (Ord(Ch)=13);
end;

```

```

Procedure Mort;
begin
    If (R<>2) and (R<>3) then T[I,J]:=0;
end;

```

```

Procedure Naissance;
begin
    If (R=3) then T[I,J]:=1;
end;

```

```

Procedure Calcul;
begin
    gen:=gen+1;
    For I:=2 to 8 do
        For J:=2 to 8 do
            begin
                R:=0;
                If T[I-1,J-1]=1 then R:=R+1;
                If T[I-1,J]=1 then R:=R+1;
                If T[I-1,J+1]=1 then R:=R+1;
                If T[I,J-1]=1 then R:=R+1;
                If T[I,J+1]=1 then R:=R+1;
                If T[I+1,J-1]=1 then R:=R+1;
                If T[I+1,J]=1 then R:=R+1;
            end;
        end;
    end;

```

```

        If T[I+1,J+1]=1 then R:=R+1;
        If (T[I,J]=1) then Mort Else Naissance;
    end;
end;
Procedure Affichage;
begin
    GotoXY(11,1);
    Writeln('Generation ',Gen);
    GotoXY(11,2);
    Writeln('-----');
    For Y:=1 to 10 do
    begin
        GotoXY(10,Y+5);
        For X:=1 to 10 do
            If (T[X,Y]=1) then Write(Chr(230),' ')
                else Write(' ');
        end;
    end;
end;

Procedure Generations;
begin
    ClrScr;
    Affichage;
    Repeat
        Calcul;
        Affichage;
        I:=0;
        Repeat
            I:=I+1;
        until (KeyPressed) or (I=40);
        If KeyPressed then Read(Kbd,Ch);
    Until (UpCase(Ch)='T');
end;

Procedure Fin;
begin
    Tot:=0;
    For I:=1 to 10 do
        For J:=1 to 10 do
            Tot:=Tot+T[I,J];
        GotoXY(1,20);
        If Tot=0 then Writeln('Population d2cimee en ',gen,' generation(s).')
            else Writeln('Fin de la simulation a la generation ',gen);
    end;
end;

```

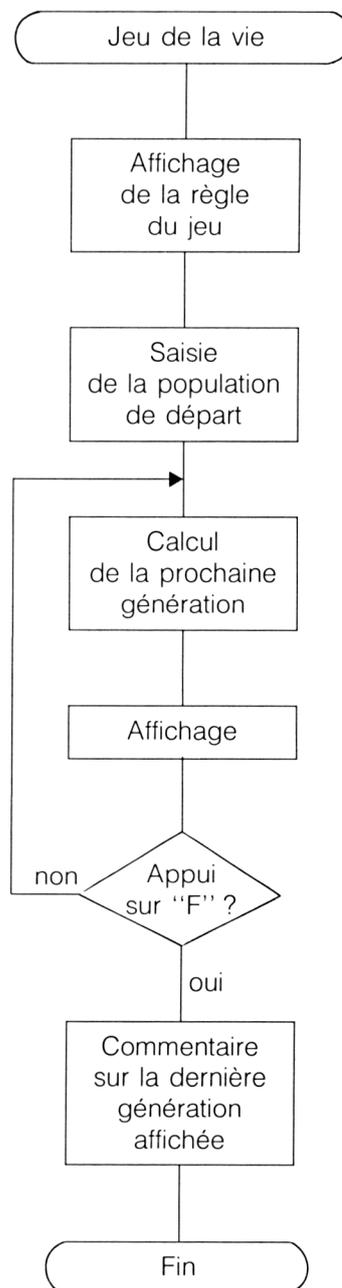
```
      {-----}  
      { PROGRAMME PRINCIPAL }  
      {-----}  
  
begin  
  Initialisation; { Init du jeu           }  
  Generations;   { Generations successives }  
  Fin;           { Fin du jeu             }  
end.
```

Quel que soit le langage choisi, lancez le programme. Les règles de survie sont affichées sur l'écran. Appuyez sur une touche de clavier. Le programme vous demande d'entrer la population de départ. Pour ce faire, utilisez les touches flèches pour déplacer le curseur. Appuyez sur <F1> pour positionner une cellule et sur <F2> pour en enlever une. Appuyez sur <Enter> lorsque la population est entrée. Les règles de survie sont appliquées à la population de départ et les générations successives sont affichées. Vous pouvez à tout moment quitter le programme en appuyant sur la touche "F".

#### LE PROGRAMME EN DÉTAIL

Les versions Basic et Turbo Pascal étant assez proches, nous nous contenterons d'analyser la version Turbo Pascal.

La logique du programme apparaît dans l'ordinogramme suivant :



Le programme débute par la section de déclaration des variables. Remarquez en particulier le tableau **T** qui contient l'état des cellules :

```
Var T : Array[1..10,1..10] of Byte ;
```

La section de déclaration des variables est suivie de la section des procédures.

La procédure **Initialisation** affiche les règles de survie sur l'écran :

```
ClrScr ;
Writeln('      Jeu de la vie') ;
Writeln('      -----') ;
...
Writeln('      Appuyez sur une touche...') ;
```

L'attente de l'appui sur une touche du clavier se fait à l'aide de la fonction **KeyPressed**. Le buffer clavier est vidé par l'instruction **Read** :

```
Repeat
Until KeyPressed
Read(Kbd,Ch) ;
```

Le tableau de jeu est ensuite mis à zéro :

```
For I: = 1 to 10 do
  For J: = 1 to 10 do
    T[I,J]: = 0 ;
```

Il est affiché sur l'écran pour que l'utilisateur puisse le remplir selon sa volonté :

```
For I: = 1 to 10 do
begin
  GotoXY(10,i + 5) ;
  For J: = 1 to 10 do
    Write ('.') ;
end ;
```

Les touches utilisables pour définir la génération de départ sont rappelées en bas de l'écran à l'aide d'instructions **Writeln** :

```
GotoXY(1,19) ;
Writeln('Déplacez le curseur à l'aide des') ;
...
Writeln('données entrées') ;
```

Une boucle **Repeat Until** permet de saisir les touches frappées et d'exécuter les actions qui doivent en découler :

```
Repeat
  While Not KeyPressed do ;
  Read(Kbd,Ch) ;
```

Une instruction **Case** permet d'identifier la touche pressée :

```
Case Ord(Ch) of
```

L'appui sur la touche flèche « vers la droite » (code ASCII 243) implique un déplacement du curseur vers la droite. Ce déplacement est possible si l'abscisse du curseur est inférieure à 10 :

```
Begin
  if (X < 10) then
    begin
```

Le caractère situé à l'ancienne position du curseur est restitué :

```
      c: = T[X,Y] ;
      GotoXY(8 + X*2, Y + 5) ;
      If (C = 0) then Write('.')
        else Write(Chr(230)) ;
```

L'abscisse est incrémentée et le curseur déplacé vers la droite :

```
      X: = X + 1 ;
      GotoXY(8 + X*2, Y + 5) ;
      Write(Chr(159)) ;
```

Des structures similaires permettent de déplacer le curseur vers la gauche, le haut et le bas :

{ Déplacement vers la gauche }

```
Begin
  If (X > 1) then
    begin
      c: = T[X,Y] ;
      GotoXY(8 + X*2, Y + 5) ;
      If (C = 0) then Write('.')
        else Write(Chr(230)) ;
      X: = X - 1 ;
      GotoXY(8 + X*2, Y + 5) ;
      Write(Chr(159)) ;
    end ;
end ;
```

{ Déplacement vers le haut }

```
Begin
  If (Y > 1) then
    begin
      c: = T[X,Y] ;
      GotoXY(8 + X*2, Y + 5) ;
      If (C = 0) then Write('.')
        else Write(Chr(230)) ;
      Y: = Y - 1 ;
      GotoXY(8 + X*2, Y + 5) ;
      Write(Chr(159)) ;
    end ;
end ;
```

```

{ Déplacement vers le bas }
Begin
  If (Y < > 10) then
  begin
    c: = T[X,Y];
    GotoXY(8 + X*2,Y + 5);
    If (C = 0) then Write('.')
      else Write(Chr(230));
    Y: = Y + 1;
    GotoXY(8 + X*2,Y + 5);
    Write(Chr(159));
  end;
end;

```

L'appui sur la touche <F1> initialise à 1 l'élément T[I,J] :

```
T[X,Y]: = 1;
```

L'appui sur la touche <F2> initialise à 0 l'élément T[I,J] :

```
T[X,Y]: = 0;
```

La boucle de saisie se termine lors de l'appui sur la touche <Enter> :

```
until (Ord(Ch) = 13);
```

Les procédures **Mort** et **Naissance** initialisent (respectivement) à zéro et à un l'élément I,J du tableau T en fonction du nombre de cellules voisines.

Toute cellule non entourée de deux ou trois autres cellules meurt :

```
If (R < > 2) and (R < > 3) then T[I,J]: = 0;
```

Toute cellule entourée de trois autres donne naissance à une nouvelle cellule :

```
If (R = 3) then T[I,J]: = 1;
```

La procédure **Calcul** analyse le tableau T, et applique les règles de survie à ce tableau. Chaque élément du tableau est analysé à l'intérieur de deux boucles imbriquées For :

```

For I: = 2 to 8 do
  For J: = 2 to 8 do
  begin

```

Pour chaque cellule, la variable R contient le nombre de cellules voisines :

```

R: = 0;
If T[I - 1,J - 1] = 1 then R: = R + 1;
If T[I - 1,J] = 1 then R: = R + 1;
If T[I - 1,J + 1] = 1 then R: = R + 1;
If T[I,J - 1] = 1 then R: = R + 1;
If T[I,J + 1] = 1 then R: = R + 1;
If T[I + 1,J - 1] = 1 then R: = R + 1;
If T[I + 1,J] = 1 then R: = R + 1;
If T[I + 1,J + 1] = 1 then R: = R + 1;

```

Lorsque l'espace pointé contient une cellule, la procédure **Mort** est activée. Dans le cas contraire, la procédure **Naissance** est activée :

```
If (T[I,J] = 1) then Mort Else Naissance;
```

Comme son nom l'indique, la procédure **Affichage** affiche sur l'écran l'état des cellules pour chaque génération.

Le numéro de la génération est tout d'abord affiché :

```
GotoXY(11,1);
Writeln('Génération ',Gen);
GotoXY(11,2);
Writeln('-----');
```

Deux boucles **For** imbriquées sont ensuite utilisées pour afficher un caractère de code ASCII 230 pour chaque cellule vivante :

```
For Y: = 1 to 10 do
begin
  GotoXY(10,Y + 5);
  For X: = 1 to 10 do
    If (T[X,Y] = 1) then Write(Chr(230),' ')
    else Write(' ');
end;
```

La procédure **Generations** est la procédure principale du programme. Elle active séquentiellement les procédures **Calcul** et **Affichage** pour afficher les générations successives **ClrScr**;

```
Affichage;
Repeat
  Calcul;
  Affichage;
```

et tester si la touche « **F** » a été pressée. A ce sujet, remarquez la boucle **Repeat Until** qui permet de capturer 40 fois consécutives l'état du clavier. Sans cette boucle, le programme aurait beaucoup de mal à détecter l'appui sur la touche **T** :

```
I: = 0;
Repeat
  I: = I + 1;
until (KeyPressed) or (I = 40);
If KeyPressed then Read(Kbd,Ch);
```

La boucle **Repeat** prend fin lorsque l'appui sur la touche « **F** » a été détecté :

```
Until (UpCase(Ch) = 'T');
```

La procédure **Fin** commente la fin du programme. Un message précise si toutes les cellules ont été décimées ou si la simulation a été arrêtée avant la décimation totale.

```
Tot: = 0;
For I: = 1 to 10 do
```

```
For J: = 1 to 10 do
  Tot: = Tot + T[I,J];
  GotoXY(1,20);
  If Tot = 0 then Writeln('Population decimee en ',gen,'
generation(s).')
  else Writeln('Fin de la simulation a la generation ',gen);
```

Le programme se termine par la section de programme principal qui active les trois procédures principales du programme :

```
Initialisation; { Init du jeu }
Generations;   { Générations successives }
Fin;           { Fin du jeu }
```