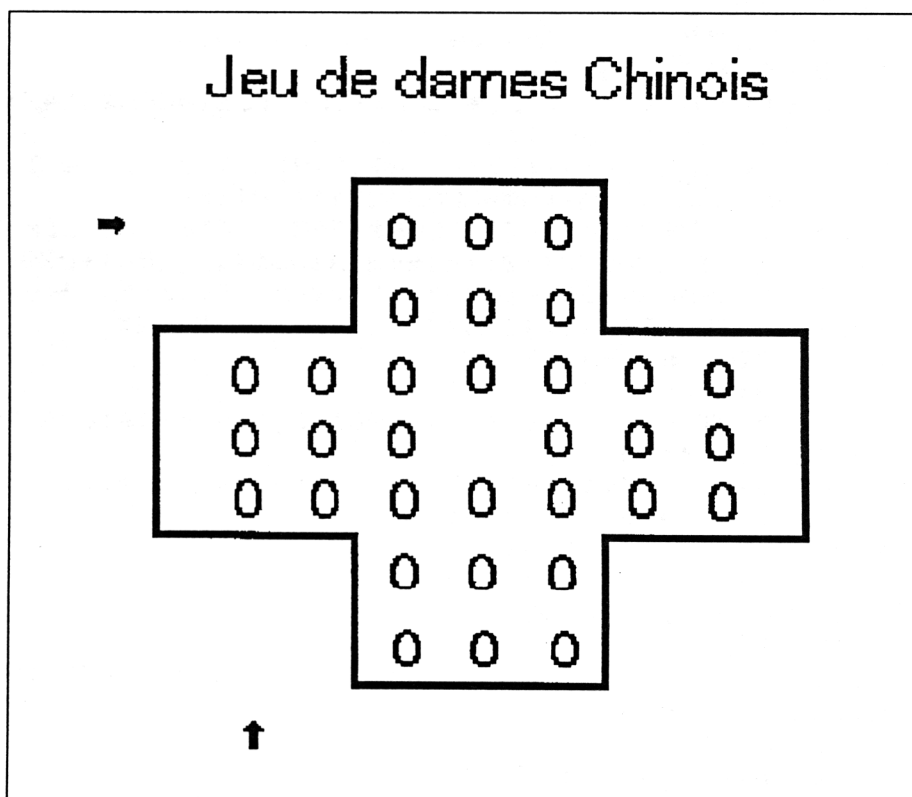


9/3.12

Jeu de dames chinois

Le jeu de dames chinois est un classique dans les jeux de réflexion. Trente-deux pions sont disposés sur une surface de jeu en forme de « plus ». Toutes les cases sont occupées sauf celle située au centre. Le but du jeu est de faire disparaître tous les pions en utilisant la règle de déplacement du jeu de dames : un pion peut sauter par dessus un autre si la case adjacente est vide.

Les pions sont donc organisés comme suit au début d'une partie :



COMMENT UTILISER LE PROGRAMME

Le programme est présenté dans deux versions : Basic et Turbo Pascal.

Voici le listing du programme Basic :

```

1000 ' =====
1010 ' Jeu de dames Chinois
1020 ' =====
1030 '
1040 GOSUB 1090 'Affichage de la grille de jeu
1050 GOSUB 1430 'Initialisation du jeu
1060 GOSUB 1590 'Deroulement d'une partie
1070 GOSUB 2590 'Commentaires sur la partie
1080 END
1090 '
1100 ' - - - - -
1110 ' Affichage de la grille de jeu
1120 ' - - - - -
1130 '
1140 MODE 1
1150 PRINT"          Jeu de dames Chinois"
1160 LOCATE 1,6
1170 c1$=CHR$(150):c2$=CHR$(156):c3$=CHR$(147):c4$=CHR$(153)
'Angles de la grille
1180 l1$=CHR$(154):l2$=CHR$(149) 'Traits de la grilles
1190 v$=CHR$(230):p$=CHR$(231) 'Pions
1200 f1$=CHR$(243):f2$=CHR$(240) 'Fleches directrices
1210 PRINT SPACE$(10)+c1$+STRING$(11,154)+c2$
1220 x$=SPACE$(10)+l2$+" "+v$+" "+v$+" "+v$+" "+l2$
1230 y$=SPACE$(10)+l2$+" "+l2$
1240 PRINT x$:PRINT y$
1250 PRINT x$
1260 PRINT " "+c1$+STRING$(5,11$)+c4$+8SPACE$(11)+c3$+STRIN
NG$(5,11$)+c2$
1270 z$=" "+l2$+" "+v$+" "+v$+" "+v$+" "+v$+" "+v$+"
"+v$+" "+v$+" "+l2$
1280 z2$=" "+l2$+SPACE$(23)+l2$
1290 PRINT z$:PRINT z2$
1300 PRINT z$:PRINT z2$
1310 PRINT z$
1320 PRINT " "+c3$+STRING$(5,11$)+c2$+SPACE$(11)+c1$+STRIN
NG$(5,11$)+c4$
1330 PRINT x$
1340 PRINT y$:PRINT x$
1350 PRINT SPACE$(10)+c3$+STRING$(11,154)+c4$
1360 LOCATE 17,13:PRINT" "
1370 LOCATE 1,7:PRINT f1$

```

```

1380 LOCATE 8,21:PRINT f2$
1390 li=1:co=1 'Position des fleches au depart
1400 ap=0 'Pas d'appui sur <ENTREE> pour l'instant
1410 RETURN
1420 '
1430 ' -----
1440 ' Initialisation du jeu
1450 ' -----
1460 '
1470 DIM t(7,7)
1480 FOR i=1 TO 7
1490   FOR j=1 TO 7
1500     t(i,j)=1
1510   NEXT j
1520 NEXT i
1530 t(1,1)=2:t(1,2)=2:t(2,1)=2:t(2,2)=2 'elements interdits
1540 t(1,6)=2:t(1,7)=2:t(2,6)=2:t(2,7)=2 'elements interdits
1550 t(6,1)=2:t(7,1)=2:t(6,2)=2:t(7,2)=2 'elements interdits
1560 t(6,6)=2:t(6,7)=2:t(7,6)=2:t(7,7)=2 'elements interdits
1570 t(4,4)=0 'case libre centrale
1580 RETURN
1590 '
1600 ' - - - - -
1610 ' Deroulement d'une partie
1620 ' - - - - -
1630 '
1640 ' - * - * - * - * - *
1650 ' Boucle de saisie
1660 ' - * - * - * - * - *
1670 '
1680 a$=INKEY$:IF a$="" THEN 1680
1690 a=ASC(a$)
1700 IF a=240 THEN GOSUB 1760 'Deplacement vers le haut
1710 IF a=241 THEN GOSUB 1850 'Deplacement vers le bas
1720 IF a=242 THEN GOSUB 1940 'Deplacement vers la gauche
1730 IF a=243 THEN GOSUB 2030 'Deplacement vers la droite
1740 IF a=13 THEN GOSUB 2120 'Selection
1750 IF UPPER$(a$)<>"Q" THEN 1680 ELSE RETURN 'Boucle de jeu
1760 '
1770 ' - * - * - * - * - * - * - *
1780 ' Deplacement vers le haut
1790 ' - * - * - * - * - * - * - *
1800 '
1810 IF li=1 THEN SOUND 1,400:GOTO 1840

```

```

1820 LOCATE 1,2*(li-1)+7:PRINT" "
1830 li=li-1:LOCATE 1,2*(li-1)+7:PRINT f1$
1840 RETURN
1850 '
1860 ' - * - * - * - * - * - * - *
1870 ' Deplacement vers le bas
1880 ' - * - * - * - * - * - * - *
1890 '
1900 IF li=7 THEN SOUND 1,400:GOTO 1930
1910 LOCATE 1,2*(li-1)+7:PRINT" "
1920 li=li+1:LOCATE 1,2*(li-1)+7:PRINT f1$
1930 RETURN
1940 '
1950 ' - * - * - * - * - * - * - *
1960 ' Deplacement vers la gauche
1970 ' - * - * - * - * - * - * - *
1980 '
1990 IF co=1 THEN SOUND 1,400:GOTO 2020
2000 LOCATE 3*(co-1)+8,21:PRINT" "
2010 co=co-1:LOCATE 3*(co-1)+8,21:PRINT f2$
2020 RETURN
2030 '
2040 ' - * - * - * - * - * - * - *
2050 ' Deplacement vers la droite
2060 ' - * - * - * - * - * - * - *
2070 '
2080 IF co=7 THEN SOUND 1,400:GOTO 2110
2090 LOCATE 3*(co-1)+8,21:PRINT" "
2100 co=co+1:LOCATE 3*(co-1)+8,21:PRINT f2$
2110 RETURN
2120 ' - * - * - *
2130 ' Selection
2140 ' - * - * - *
2150 '
2160 IF t(li,co)=2 THEN SOUND 1,400:GOTO 2220 'Erreur dans l
a selection
2170 IF (ap=0) AND t(li,co)=1 THEN GOSUB 2230:GOTO 2220 'Mem
orisation du coup
2180 IF (ap=1) AND t(li,co)=1 THEN SOUND 1,400:GOTO 2220 'Er
reur dans la selection
2190 IF (ap=1) AND t(li,co)=3 THEN GOSUB 2510:GOTO 2220 'Ann
ulation
2200 IF (ap=1) AND ((co+2=cm) OR (co-2=cm) OR (lm+2=li) OR (
lm-2=li)) THEN 2210 ELSE SOUND 1,400:RETURN
2210 IF (ap=1) AND t(li,co)=0 THEN GOSUB 2330:GOTO 2220 'Dep
lacement
2220 RETURN
2230 '

```

```

2240 ' - * - * - * - * - * - *
2250 ' Memorisation du coup
2260 ' - * - * - * - * - * - *
2270 '
2280 LOCATE (co-1)*3+8,(li-1)*2+7:PRINT p$
2290 t(li,co)=3
2300 lm=li:cm=co 'Memorisation
2310 ap=1 '1er pion positionne
2320 RETURN
2330 '
2340 ' - * - * - * - *
2350 ' Deplacement
2360 ' - * - * - * - *
2370 '
2380 cas=0
2390 IF (lm=li) AND (co+2=cm) THEN cas=1
2400 IF (lm=li) AND (co-2=cm) THEN cas=2
2410 IF (lm+2=li) AND (co=cm) THEN cas=3
2420 IF (lm-2=li) AND (co=cm) THEN cas=4
2430 IF cas=0 THEN SOUND 1,400:RETURN
2440 IF cas=1 THEN LOCATE (cm-2)*3+8,(lm-1)*2+7:PRINT "
":t(lm,cm)=0:t(lm,cm-1)=0:t(li,co)=1
2450 IF cas=2 THEN LOCATE (cm-1)*3+8,(lm-1)*2+7:PRINT "
":t(lm,cm)=0:t(lm,cm+1)=0:t(li,co)=1
2460 IF cas=3 THEN LOCATE (cm-1)*3+8,(lm-1)*2+7:PRINT " ":LO
CATE (cm-1)*3+8,lm*2+7:PRINT " ":t(lm,cm)=0:t(lm+1,cm)=0:t(l
i,co)=1
2470 IF cas=4 THEN LOCATE (cm-1)*3+8,(lm-1)*2+7:PRINT " ":LO
CATE (cm-1)*3+8,(lm-2)*2+7:PRINT " ":t(lm,cm)=0:t(lm-1,cm)=0
:t(li,co)=1
2480 IF cas=0 THEN SOUND 1,400 ELSE ap=0:LOCATE (co-1)*3+8,(
li-1)*2+7:PRINT v$
2490 ap=0 ' coup joue
2500 RETURN
2510 ' - * - * - * - *
2520 ' Annulation
2530 ' - * - * - * - *
2540 '
2550 LOCATE (co-1)*3+8,(li-1)*2+7:PRINT v$
2560 t(li,co)=1
2570 ap=0 ' coup joue
2580 RETURN
2590 ' -----
2600 ' Commentaires sur la partie
2610 ' -----
2620 '
2630 FOR i=1 TO 7

```



```

Writeln(z);
Writeln(z2);
Writeln(z);
Writeln(z2);
Writeln(z);
Writeln('      '+c3+l1+l1+l1+l1+l1+c2+'          '+c1+l1+l1+l1+l1+l1+c4);
Writeln(x);
Writeln(y);
Writeln(x);
Writeln('          '+c3+l1+l1+l1+l1+l1+l1+l1+l1+l1+l1+l1+c4);
GotoXY(17,13);
Write(' ');
GotoXY(1,7);
Writeln(f1);
GotoXY(8,21);
Writeln(f2);
li:=1;
  co:=1;
  ap:=0;
end;

```

```

Procedure Init;
{ Initialisation du jeu }
begin
  FOR i:=1 TO 7 do
    FOR j:=1 TO 7 do
      t[i,j]:=1;
      t[1,1]:=2; t[1,2]:=2; t[2,1]:=2; t[2,2]:=2;
      t[1,6]:=2; t[1,7]:=2; t[2,6]:=2; t[2,7]:=2;
      t[6,1]:=2; t[7,1]:=2; t[6,2]:=2; t[7,2]:=2;
      t[6,6]:=2; t[6,7]:=2; t[7,6]:=2; t[7,7]:=2;
      t[4,4]:=0;
end;

```

```

Procedure Haut;
{ Deplacement du curseur vers le haut }
begin
  IF li<>1 THEN
    begin
      GotoXY(1,2*(li-1)+7);
      Writeln(' ');
      li:=li-1;
      GotoXY(1,2*(li-1)+7);
      Writeln(f1);
    end;
end;

```

```
Procédure Bas;  
{ Déplacement du curseur vers le bas }  
begin  
  IF li<>7 THEN  
    begin  
      GotoXY(1,2*(li-1)+7);  
      Writeln(' ');  
      li:=li+1;  
      GotoXY(1,2*(li-1)+7);  
      Writeln(f1);  
    end;  
end;
```

```
Procédure Gauche;  
{ Déplacement du curseur vers la gauche }  
begin  
  IF co<>1 THEN  
    begin  
      GotoXY(3*(co-1)+8,21);  
      Writeln(' ');  
      co:=co-1;  
      GotoXY(3*(co-1)+8,21);  
      Writeln(f2);  
    end;  
end;
```

```
Procédure Droite;  
{ Déplacement du curseur vers la droite }  
begin  
  IF co<>7 THEN  
    begin  
      GotoXY(3*(co-1)+8,21);  
      Writeln(' ');  
      co:=co+1;  
      GotoXY(3*(co-1)+8,21);  
      Writeln(f2);  
    end;  
end;
```

```
Procédure Memorisation;  
{ Memorisation d'un pion selectionne }  
begin  
  GotoXY((co-1)*3+8,(li-1)*2+7);  
  Writeln(p);  
end;
```



```

t[li,col]:=3;
lm:=li;
cm:=co;
ap:=1;
end;

Procedure Deplacement;
( Deplacement d'un pion )
begin
  cas:=0;
  IF (lm=li) AND (co+2=cm) THEN cas:=1;
  IF (lm=li) AND (co-2=cm) THEN cas:=2;
  IF (lm+2=li) AND (co=cm) THEN cas:=3;
  IF (lm-2=li) AND (co=cm) THEN cas:=4;

  IF cas=1 THEN
  begin
    GotoXY((cm-2)*3+8,(lm-1)*2+7);
    Writeln(' ');
    t[lm,cm]:=0;
    t[lm,cm-1]:=0;
    t[li,col]:=1;
  end;

  IF cas=2 THEN
  begin
    GotoXY((cm-1)*3+8,(lm-1)*2+7);
    Writeln(' ');
    t[lm,cm]:=0;
    t[lm,cm+1]:=0;
    t[li,col]:=1;
  end;

  IF cas=3 THEN
  begin
    GotoXY((cm-1)*3+8,(lm-1)*2+7);
    Writeln(' ');
    GotoXY((cm-1)*3+8,lm*2+7);
    Writeln(' ');
    t[lm,cm]:=0;
    t[lm+1,cm]:=0;
  end;

  IF cas=4 THEN
  begin

```

```

GotoXY((cm-1)*3+8, (lm-1)*2+7);
Writeln(' ');
GotoXY((cm-1)*3+8, (lm-2)*2+7);
Writeln(' ');
t[lm,cm]:=0;
t[lm-1,cm]:=0;
t[li,col]:=1;
end;

IF cas<>0 THEN
begin
  ap:=0;
  GotoXY((co-1)*3+8, (li-1)*2+7);
  Writeln(v);
end;
ap:=0;
end;

Procedure Annulation;
{ Annulation d'une selection }
begin
  GotoXY((co-1)*3+8, (li-1)*2+7);
  Writeln(v);
  t[li,col]:=1;
  ap:=0;
end;

Procedure Selection;
{ Selection d'un pion }
begin
  IF (ap=0) AND (t[li,col]=1)
  THEN Memorisation
  ELSE
  IF (ap=1) AND (t[li,col]=3)
  THEN Annulation
  ELSE
  IF (ap=1) AND ((co+2=cm) OR (co-2=cm) OR (lm+2=li) OR (lm-2=li)) THEN
  IF (ap=1) AND (t[li,col]=0) THEN Deplacement;
end;

Procedure Partie;
{ Deroulement d'une partie }
begin
  Repeat
  While Not KeyPressed do;

```

```

    Read(Kbd,a);
    IF a=#240 THEN Haut;
    IF a=#241 THEN Bas;
    IF a=#242 THEN Gauche;
    IF a=#243 THEN Droite;
    IF a=#13 THEN Selection;
  Until Ucase(a)='Q';
end;

Procedure Commente;
{ Commentaires sur la partie }
begin
  pion:=0;
  FOR i:=1 TO 7 do
    FOR j:=1 TO 7 do
      IF t[i,j]=1 THEN pion:=pion+1;
    ClrScr;
    IF pion=1 THEN Writeln('BRAVO');
    IF pion<>1 THEN Writeln( pion,' pions en jeu. Perseverez. ');
  end;

                                {-----}
                                { Programme principal }
                                {-----}

begin
  Affiche;   { Affichage de la grille de jeu }
  Init;     { Initialisation du jeu           }
  Partie;   { Deroulement d' une partie     }
  Commente; { Commentaires sur la partie     }
end.

```

Ces deux programmes ont un comportement tout à fait semblable si ce n'est un point de détail : le programme Basic signale toute mauvaise manipulation par un bip sonore, alors que le programme Turbo Pascal se contente de l'ignorer.

Lancez le programme. La grille de jeu apparaît sur l'écran, sur 40 colonnes pour le programme Basic et sur 80 colonnes pour le programme Turbo Pascal.

Utilisez les touches flèches du clavier pour que l'intersection entre la flèche horizontale et la flèche verticale se trouve sur le pion à sélectionner. Appuyez alors sur la touche <ENTER>. Le pion sélectionné change de couleur. Utilisez à nouveau les touches flèches du clavier pour pointer une case vide située à deux cases du pion sélectionné. Appuyez sur <ENTER>. Le pion sélectionné est déplacé sur la case

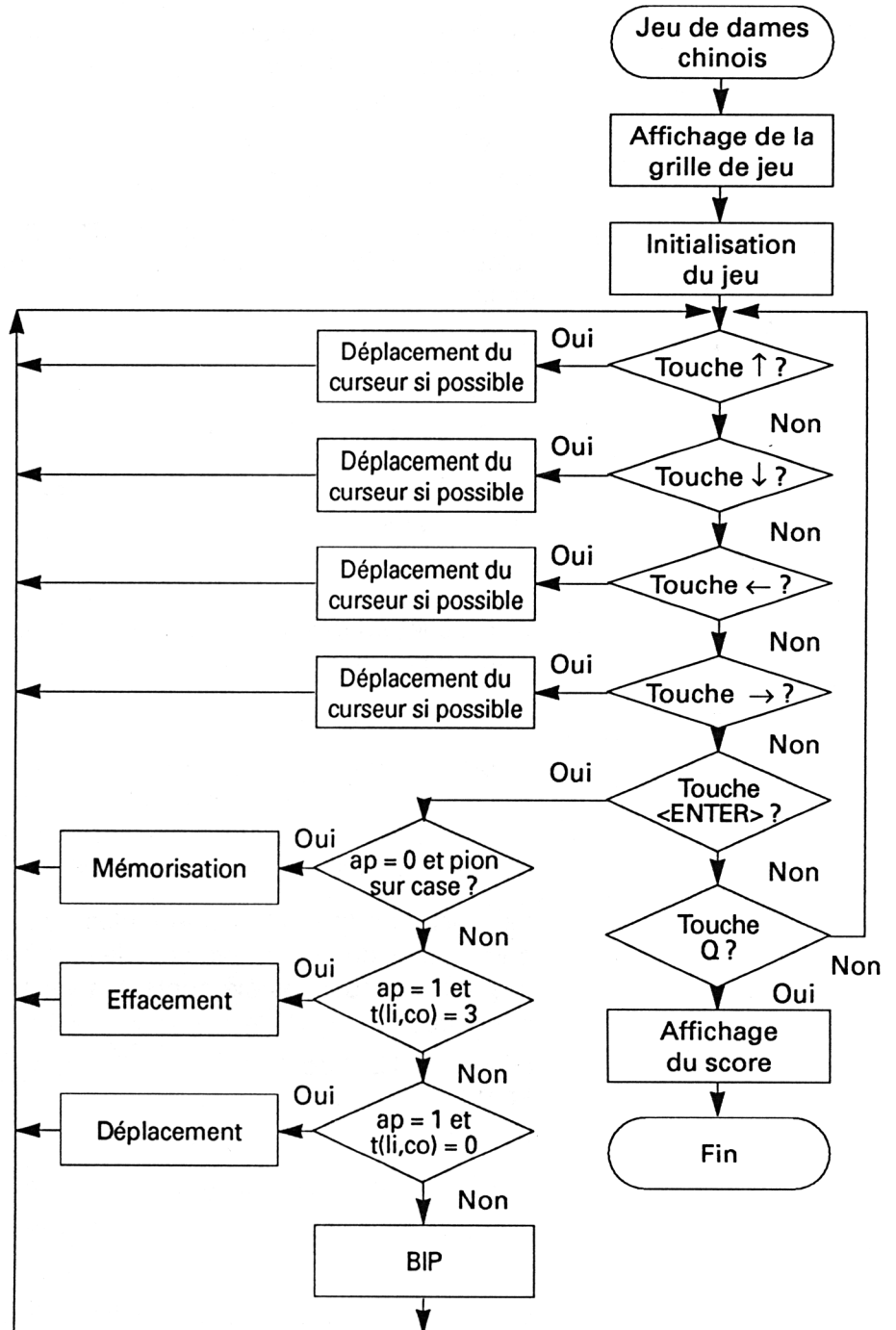
vide, et le pion sauté est effacé. Recommencez ce processus pour éliminer le plus de pions possible.

Vous pouvez quitter à tout moment la partie en appuyant sur la touche **Q** (comme **Q**uitter).

Vous pouvez également désélectionner un pion sélectionné en appuyant sur **<ENTER>** après avoir convenablement positionné les flèches.

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Nous allons analyser en détail le fonctionnement du programme Turbo Pascal. Vous n'aurez aucun mal à transférer les explications au programme Basic.

Le programme commence par la section des variables.

De nombreuses variables (caractère et chaîne) sont utilisées pour stocker les caractères semi-graphiques et les séquences qui permettront de représenter la grille de jeu :

```
c1, c2, c3, c4, 11, 12, v, p, f1, f2, a : Char;  
x, y, z, z2 : String [35];
```

Quelques variables de type « octet » permettent d'effectuer divers calculs dans le programme, comme par exemple le nombre de pions en jeu ou encore les coordonnées de la case sélectionnée :

```
li, co, ap, i, j, cm, lm, pion, cas : Byte;
```

Le tableau de jeu a pour nom T :

```
T : Array [1.. 7, 1.. 7] of Byte;
```

Il contient des informations numériques qui renseignent sur la disposition des pions :

```
0 : case vide,  
1 : case occupée par un pion,  
2 : case interdite (car n'appartenant pas au jeu),  
3 : case occupée par un pion sélectionné.
```

Le programme se poursuit par la section des procédures.

La procédure **Affiche** est responsable de l'affichage du jeu au départ. Elle utilise les variables caractère et chaîne dont nous avons parlé pour calculer les éléments graphiques de la grille de jeu. Ces éléments sont affichés par des instructions **Writeln**.

La procédure **Init** initialise le tableau de jeu en respectant les codes numériques dont nous avons parlé plus haut. Dans un premier temps, elle initialise tous les éléments à la valeur 1 à l'aide de deux boucles imbriquées :

```
FOR i:= 1 TO 7 do  
FOR j:=1 TO 7 do  
t[i, j] :=1 ;
```

Elle corrige ensuite certaines de ces valeurs pour :

- interdire les cases situées sur les côtés,
- vider la case centrale.

Les procédures **Haut**, **Bas**, **Gauche** et **Droite** déplacent le « curseur » (c'est-à-dire des flèches) dans une des quatre directions. Le déplacement n'est effectué que s'il est possible.

Prenons par exemple le cas d'un déplacement vers le haut. L'appui sur la touche flèche « vers le haut » provoquera un déplacement de la flèche horizontale d'une position vers le haut si et seulement si sa position actuelle le permet. Elle ne doit donc pas faire référence à la première ligne du jeu :

```
IF li <> 1 then
```

La flèche horizontale est effacée. L'indicateur de position verticale est décrémenté, et la flèche horizontale est affichée à sa nouvelle position :

```
GotoXY (1, 2*(li-1)+7) ;
Writeln ( ' ' ) ;
li:=li-1 ;
GotoXY (1, 2*(li-1)+7) ;
Writeln (f1) ;
```

La procédure **Memorisation** est activée lors de la sélection d'un pion. Elle modifie la couleur du pion :

```
GotoXY ((co-1)*3+8, (li-1)*2+7) ;
Writeln (p) ;
```

et initialise la case correspondante à la valeur 3 dans le tableau T :

```
t [li, co]:=3 ;
```

Les coordonnées du pion sélectionné sont stockées dans les variables **lm** et **cm**, et la variable **ap** est initialisée à 1 pour indiquer qu'un pion a été sélectionné.

La procédure **Deplacement** est activée lorsque le joueur sélectionne une case vide alors qu'un pion a déjà été sélectionné.

Les configurations autorisées sont les suivantes :

```
IF (lm=li) AND (co+2=cm) THEN CAS:=1;
IF (lm=li) AND (co-2=cm) THEN CAS:=2;
IF (lm+2=li) AND (co=cm) THEN CAS:=3;
IF (lm-2=li) AND (co=cm) THEN CAS:=4;
```

Lorsqu'un de ces cas est détecté, le pion sélectionné est déplacé, et le pion sauté est effacé. Toutes ces modifications sont mémorisées dans le tableau T :

Par exemple pour le cas 1 :

– effacement du pion sélectionné et du pion sauté :

```
GotoXY ((cm-2)*3+8, (lm-1)*2+7) ;
Writeln ( ' ' ) ;
```

– mémorisation des changements dans le tableau T :

```
t [lm, cm] :=0; --> effacement du pion sélectionné
```

```
t [lm, cm-1] :=0; --> effacement du pion sauté
t [li, co] :=1;    --> mémorisation de la nouvelle position du pion
– affichage du pion à sa nouvelle position :
GotoXY ((co-1)*3+8, (li-1)*2+7) ;
Writeln (v) ;
```

La procédure **Annulation** est activée lorsque le joueur sélectionne un pion déjà sélectionné.

```
La couleur du pion est restituée :
GotoXY ((co-1)*3+8, (li-1)*2+7) ;
Writeln (v) ;
```

La restitution est mémorisée dans le tableau T :

```
t [li, co] :=1 ;
```

et la variable **ap** est initialisée à 0 pour indiquer qu'aucun pion n'est actuellement sélectionné :

```
ap:=0;
```

La procédure **Selection** est activée à chaque appui sur la touche <ENTER>.

Lorsqu'aucun pion n'a été préalablement sélectionné, et lorsque la case courante contient un pion, la procédure **Memorisation** est activée :

```
IF (ap=0) AND (t [li, co]=1)
THEN Memorisation ;
```

Lorsqu'un pion a été préalablement sélectionné, et lorsque la case courante contient la valeur 3, le joueur désire refaire sa sélection. La procédure **Annulation** est donc activée :

```
IF (ap=1) AND (t [li, co]=3)
THEN Annulation ;
```

Lorsqu'aucun de ces cas n'est vérifié, le joueur désire certainement effectuer un déplacement. Le déplacement sera effectué si :

- un pion a été sélectionné,
- la case courante est vide et se trouve à une distance « convenable » du pion sélectionné :

```
IF (ap=1) AND ((co+2=cm) OR (co-2=cm) OR (lm+2=li) OR (lm-2=li))
THEN
IF (ap=1) AND (t [li, co]=0) THEN Deplacement;
```

La procédure **Partie** est responsable du déroulement d'une partie. C'est donc la procédure principale du programme. Elle est composée d'une boucle **Repeat Until** qui prend fin lorsque la touche **Q** du clavier est pressée :

Repeat

...

Until Uppcase (a) = 'Q' ;

En fonction de la touche pressée (touche flèche ou <ENTER>), Partie active une procédure de déplacement ou de sélection :

IF a=#240 THEN Haut;

IF a=#241 THEN Bas;

IF a=#242 THEN Gauche;

IF a=#243 THEN Droite;

IF a=#13 THEN Selection;

La procédure Commente est activée en fin de partie, c'est-à-dire lorsque le joueur a appuyé sur la touche Q du clavier. Cette procédure affiche un message qui dépend du nombre de pions en jeu.

Le nombre de pions est comptabilisé à l'aide de deux boucles For imbriquées :

pion :=0 ;

FOR i:=1 TO 7 do

FOR j:=1 TO 7 do

IF t [i, j]=1 THEN pion:=pion+1 ;

Le but du jeu étant d'effacer le plus de pions possible, le joueur a droit à toute la considération de la machine s'il ne lui reste plus qu'un pion :

IF pion=1 THEN Writeln ('BRAVO') ;

Un message d'encouragement l'invite à recommencer une autre partie pour améliorer son score dans le cas contraire :

IF pion <>1 THEN Writeln (pion, 'pions en jeu. Perseverez.') ;

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

- Affiche et Init pour initialiser une partie ;
- Partie pour gérer une partie ;
- Commente pour commenter la partie.

