

9/3.14

Jeu du perplexus

Le perplexus est un jeu de réflexion intéressant qui oppose deux joueurs. Pour chaque joueur, le but du jeu est d'aligner trois pions de sa couleur.

Ce jeu se joue sur six barres horizontales de six cases chacune. Un mini-langage permet de placer un pion ou de déplacer une barre horizontale vers la droite ou vers la gauche.

Ce programme est proposé en version Basic et Turbo Pascal.


```

1500 RETURN
1510 '
1520 REM -----
1530 REM Deroulement d'une partie
1540 REM -----
1550 '
1560 IF j=1 THEN j$=j2$ : j=2 ELSE j$=j1$ : j=1
1570 LOCATE 1,24
1580 PRINT "La main est au joueur ";j$
1590 INPUT "Quel est votre jeu ? ",jj$
1600 jj$=UPPER$(jj$) 'Mise en majuscule
1610 ok=0 'Commande invalide jusqu'a preuve du contraire
1620 IF MID$(jj$,1,1)="G" THEN GOSUB 1710 'Deplacement vers
la gauche
1630 IF MID$(jj$,1,1)="D" THEN GOSUB 1890 'Deplacement vers
la droite
1640 IF MID$(jj$,1,1)="L" AND MID$(jj$,3,1)="C" THEN GOSUB 2
070 'Placement
1650 IF ok=1 THEN GOSUB 2540 ELSE GOSUB 2140 'Affichage ou m
auvaise commande
1660 IF ok=0 THEN IF j=1 THEN j$=j2$ : j=2 ELSE j$=j1$ : j=1

1670 GOSUB 2260 'Test de la fin du jeu
1680 IF g=0 THEN 1520 'Boucle de jeu
1690 RETURN
1700 '
1710 REM - - - - -
1720 REM Deplacement vers la gauche
1730 REM - - - - -
1740 '
1750 l=VAL(MID$(jj$,2,1))
1760 IF l<7 AND l>0 THEN 1850
1770 PEN 2
1780 LOCATE 1,23
1790 PRINT"Commande invalide"
1800 FOR i=1 TO 1000:NEXT i
1810 LOCATE 1,23
1820 PRINT SPACE$(20)
1830 PEN 1
1840 RETURN
1850 d(1)=d(1)-1
1860 ok=1
1870 RETURN
1880 '
1890 REM - - - - -
1900 REM Deplacement vers la droite
1910 REM - - - - -
1920 '
1930 l=VAL(MID$(jj$,2,1))
1940 IF l<7 AND l>0 THEN 2030

```

```
1950 PEN 2
1960 LOCATE 1,23
1970 PRINT"Commande invalide"
1980 FOR i=1 TO 1000:NEXT i
1990 LOCATE 1,23
2000 PRINT SPACE$(20)
2010 PEN 1
2020 RETURN
2030 d(1)=d(1)+1
2040 ok=1
2050 RETURN
2060 '
2070 REM - - - - -
2080 REM Placement d'un pion
2090 REM - - - - -
2100 '
2110 l=VAL(MID$(jj$,2,1)) 'Ligne du placement
2120 c=VAL(MID$(jj$,4,1)) 'Colonne du placement
2130 IF (l>0) AND (l<7) AND (c>0) AND (c<7) AND (t(l,c)=0) T
HEN 2220
2140 PEN 2
2150 LOCATE 1,23
2160 PRINT"Commande invalide"
2170 FOR i=1 TO 1000:NEXT i
2180 LOCATE 1,23
2190 PRINT SPACE$(20)
2200 PEN 1
2210 RETURN
2220 t(l,c)=j
2230 ok=1
2240 RETURN
2250 '
2260 REM -----
2270 REM Test de fin de jeu
2280 REM -----
2290 '
2300 FOR i=1 TO 6
2310   FOR k=1 TO 18
2320     u(i,k)=0
2330   NEXT k
2340 NEXT i
2350 FOR i=1 TO 6
2360   FOR k=1 TO 6
2370     u(i,k+6+d(i))=t(i,k)
2380   NEXT k
2390 NEXT i
2400 FOR i=1 TO 6
2410   FOR k=1 TO 16
2420     IF (u(i,k)=1) AND (u(i,k+1)=1) AND (u(i,k+2)=1) THE
N g=1
```

```

2430     IF (u(i,k)=2) AND (u(i,k+1)=2) AND (u(i,k+2)=2) THE
N g=2
2440     NEXT k
2450 NEXT i
2460 FOR i=1 TO 4
2470     FOR k=1 TO 18
2480         IF (u(i,k)=1) AND (u(i+1,k)=1) AND (u(i+2,k)=1) THE
N g=1
2490         IF (u(i,k)=2) AND (u(i+1,k)=2) AND (u(i+2,k)=2) THE
N g=2
2500     NEXT k
2510 NEXT i
2520 RETURN
2530 '
2540 REM - - - - -
2550 REM Affichage du nouvel ecran de jeu
2560 REM - - - - -
2570 '
2580 CLS
2590 LOCATE 11,1
2600 PRINT"P E R P L E X U S"
2610 FOR i=1 TO 6
2620     LOCATE 13+d(i)*2,1+i*3 : PRINT s2$
2630     LOCATE 13+d(i)*2,2+i*3 : PRINT s3$
2640     LOCATE 13+d(i)*2,3+i*3 : PRINT s4$
2650 NEXT i
2660 FOR i=1 TO 6
2670     FOR k=1 TO 6
2680         IF t(i,k)=0 THEN 2710
2690         LOCATE 13+d(i)*2+k*2-1,2+i*3
2700         IF t(i,k)=1 THEN PRINT j1$ ELSE PRINT j2$
2710     NEXT k
2720 NEXT i
2730 RETURN
2740 '
2750 REM -----
2760 REM Felicitations au gagnant
2770 REM -----
2780 '
2790 CLS
2800 PRINT"Felicitations, joueur ";
2810 IF g=1 THEN PRINT j1$ ELSE PRINT j2$
2820 PRINT"Laissez une seconde chance au joueur ";
2830 IF g=1 THEN PRINT j2$ ELSE PRINT j1$
2840 RETURN

```

Si vous préférez la version Turbo Pascal, saisissez le listing suivant :

```

Program Perplexus;
{-----}
{ JEU DU PERPLEXUS }
{-----}

VAR
  c1, c2, c3, c4,
  c5, c6, c7, c8 : Char;
  s1, s2, s3, s4 : String[20];
  jd, j1, j2     : Char;
  d              : Array[1..6] of Integer;
  t              : Array[1..6,1..6] of Byte;
  u              : Array[1..6,1..18] of Byte;
  i, j, k        : Byte;
  jj             : String[4];
  ok             : Byte;
  l, c           : Integer;
  Erreur         : Integer;
  g              : Byte;

Procedure Bad;
{ ----- }
{ Erreur dans la commande }
{ ----- }

begin
  GotoXY(1,22);
  Write('Commande invalide');
  FOR i:=1 TO 10000 do;
  GotoXY(1,22);
  Write('          ');
end;

Procedure Gauche;
{ - - - - - }
{ Deplacement vers la gauche }
{ - - - - - }

begin
  VAL(Copy(jj,2,1),l,Erreur);
  IF (l<7) AND (l>0) THEN
  begin
    d[l]:=d[l]-1;
    ok:=1;
  end
  else Bad;

```

```
end;
```

```
Procedure Droite;
```

```
{ - - - - - }
{ Deplacement vers la droite }
{ - - - - - }
```

```
begin
```

```
  VAL(Copy(jj,2,1),1,Erreur);
```

```
  IF (l<7) AND (l>0) THEN
```

```
    begin
```

```
      d[l]:=d[l]+1;
```

```
      ok:=1;
```

```
    end
```

```
    else Bad;
```

```
end;
```

```
Procedure Placement;
```

```
{ - - - - - }
{ Placement d'un pion }
{ - - - - - }
```

```
begin
```

```
  VAL(Copy(jj,2,1),1,Erreur); { Ligne du placement }
```

```
  VAL(Copy(jj,4,1),c,Erreur); { Colonne du placement }
```

```
  IF (l>0) AND (l<7) AND (c>0) AND (c<7) AND (t[l,c]=0) THEN
```

```
    begin
```

```
      t[l,c]:=j;
```

```
      ok:=1;
```

```
    end
```

```
    else Bad;
```

```
end;
```

```
Procedure Test_Fin;
```

```
{ ----- }
{ Test de fin de jeu }
{ ----- }
```

```
begin
```

```
  FOR i:=1 TO 6 do
```

```
    FOR k:=1 TO 18 do
```

```
      u[i,k]:=0;
```

```
  FOR i:=1 TO 6 do
```

```

FOR k:=1 TO 6 do
  u[i,k+6+d[i]]:=t[i,k];

FOR i:=1 TO 6 do
  FOR k:=1 TO 16 do
  begin
    IF (u[i,k]=1) AND (u[i,k+1]=1) AND (u[i,k+2]=1) THEN g:=1;
    IF (u[i,k]=2) AND (u[i,k+1]=2) AND (u[i,k+2]=2) THEN g:=2;
  end;

FOR i:=1 TO 4 do
  FOR k:=1 TO 18 do
  begin
    IF (u[i,k]=1) AND (u[i+1,k]=1) AND (u[i+2,k]=1) THEN g:=1;
    IF (u[i,k]=2) AND (u[i+1,k]=2) AND (u[i+2,k]=2) THEN g:=2;
  end;
end;

```

```

Procedure Affiche;

```

```

{ - - - - - }
{ Affichage du nouvel ecran de jeu }
{ - - - - - }

```

```

begin

```

```

  ClrScr;

```

```

  GotoXY(11,1);

```

```

  Write('P E R P L E X U S');

```

```

  FOR i:=1 TO 6 do

```

```

  begin

```

```

    GotoXY(13+d[i]*2,1+i*3); Write(s2);

```

```

    GotoXY(13+d[i]*2,2+i*3); Write(s3);

```

```

    GotoXY(13+d[i]*2,3+i*3); Write(s4);

```

```

  end;

```

```

  FOR i:=1 TO 6 do

```

```

    FOR k:=1 TO 6 do

```

```

    begin

```

```

      IF (t[i,k]<>0) THEN

```

```

        begin

```

```

          GotoXY(13+d[i]*2+k*2-1,2+i*3);

```

```

          IF t[i,k]=1 THEN Write(j1) ELSE Write(j2);

```

```

        end;

```

```

    end;

```

```

end;

```

```

Procedure Init;

```

```

{ ----- }

```

```

{ Affichage de l'ecran de depart }
{ ----- }

begin
  { - - - - - }
  { Defition de l'aire de jeu }
  { - - - - - }
  ClrScr;
  c1:=CHR(150); c2:=CHR(154); c3:=CHR(158);
  c4:=CHR(156); c5:=CHR(149); c6:=CHR(147);
  c7:=CHR(155); c8:=CHR(153);
  s1:=c2+c3;
  s2:=c1+s1+s1+s1+s1+s1+c2+c4;
  s1:=c5+' ';
  s3:=s1+s1+s1+s1+s1+s1+s1;
  s1:=c2+c7;
  s4:=c6+s1+s1+s1+s1+s1+c2+c8;
  j1:=CHR(227); { Joueur 1 }
  j2:=CHR(231); { Joueur 2 }
  FOR i:=1 TO 6 do
  begin
    d[i]:=0;
    FOR j:=1 TO 6 do
      t[i,j]:=0;
    end;
  j:=2;

  { - - - - - }
  { Affichage de l'ecran }
  { - - - - - }

  GotoXY(11,1);
  Writeln('P E R P L E X U S');
  FOR i:=1 TO 6 do
  begin
    GotoXY(13,1+i*3); Write(s2);
    GotoXY(13,2+i*3); Write(s3);
    GotoXY(13,3+i*3); Write(s4);
  end;
end;

Procedure Jeu;
{ ----- }
{ Deroulement d'une partie }
{ ----- }

begin
  g:=0;

```

```

Repeat
  IF (j=1) THEN
  begin
    jd:=j2;
    j:=2;
  end
  ELSE
  begin
    jd:=j1;
    j:=1;
  end;
  GotoXY(1,23);
  Writeln('La main est au joueur ',j);
  Write('Quel est votre jeu ? ');
  Readln(jj);
  FOR k:=1 to Length(jj) do
    jj[k]:=UpCase(jj[k]);
  ok:=0;
  IF (Copy(jj,1,1)='G') THEN Gauche;
  IF (Copy(jj,1,1)='D') THEN Droite;
  IF (Copy(jj,1,1)='L') AND (Copy(jj,3,1)='C') THEN Placement;
  IF (ok=1) THEN Affiche ELSE Bad;
  IF (ok=0) THEN IF (j=1) THEN begin
                                jd:=j2;
                                j:=2;
                              end
                              ELSE
                              begin
                                jd:=j1;
                                j:=1;
                              end;
  Test_Fin; { Test de la fin du jeu }
until (g<>0);
end;

```

```

Procedure Felicite;

```

```

{ ----- }
{ Felicitations au gagnant }
{ ----- }

```

```

begin

```

```

  ClrScr;
  Write('Felicitations, joueur ');
  IF g=1 THEN Writeln(j1) ELSE Writeln(j2);
  Write('Laissez une seconde chance au joueur ');
  IF g=1 THEN Writeln(j2) ELSE Writeln(j1);
end;

```

```

{-----}
{ PROGRAMME PRINCIPAL }
{-----}

begin
  Init;      { Affichage de l'ecran de depart }
  Jeu;      { Deroulement d'une partie      }
  Felicite; { Felicitations au gagnant      }
end.

```

Lancez le programme. L'écran se présente alors comme suit :

PERPLEXUS

La main est au joueur 0.
Quel est votre jeu ?

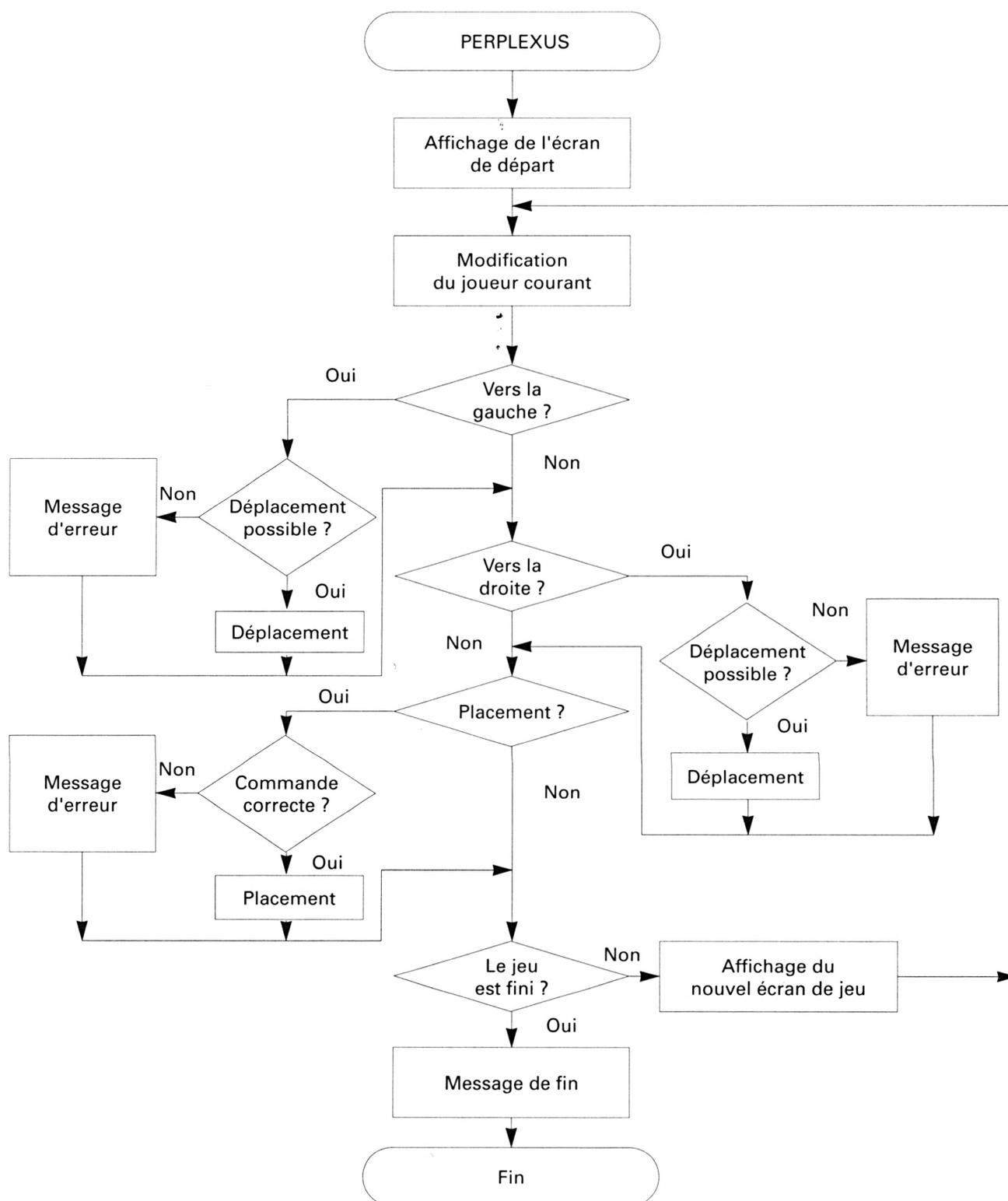
Les commandes possibles sont les suivantes :

- Dx** : déplacement de la barre horizontale **x** d'une case vers la droite.
- Gx** : déplacement de la barre horizontale **x** d'une case vers la gauche.
- LxCy** : placement d'un pion sur la case ligne **x** colonne **y**.

Lorsque le premier joueur a entré sa commande, le tour de jeu change. Le second joueur doit entrer sa commande. Ainsi de suite jusqu'à ce que trois pions de même couleur soient alignés. Le programme détecte automatiquement l'alignement horizontal ou vertical de trois pions de même couleur. Il arrête alors la partie et félicite le vainqueur.

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Seul le programme écrit en Basic sera analysé. Vous transférerez sans problème les indications données au programme Turbo Pascal.

Le programme principal se trouve dans les premières lignes du programme. Il active séquentiellement les trois sous-programmes principaux :

```
1040 GOSUB 1080 'Affichage de l'écran de départ
1050 GOSUB 1520 'Jeu
1060 GOSUB 2750 'Félicitations
```

Le premier sous-programme affiche l'écran de départ et définit toutes les variables d'ordre général utilisées par les autres sous-programmes.

Les caractères graphiques utilisés pour afficher les six barres sont définis lignes 1160 à 1180. Ils sont stockés dans les chaînes **c1\$** à **c8\$** pour faciliter leur manipulation.

```
1160 c1$=CHR$(150) : c2$=CHR$(154) : c3$=CHR$(158)
...
1180 c7$=CHR$(155) : c8$=CHR$(153)
```

Chaque barre est définie sur trois lignes. Ces trois lignes sont calculées à partir des caractères **c1\$** à **c8\$** lignes 1190 à 1240 :

```
1190 s1$=c2$+c3$
...
1240 s4$=c6$+s1$+s1$+s1$+s1$+s1$+c2$+c8$
```

Les pions des deux joueurs sont également représentés à l'aide de caractères graphiques :

```
1250 j1$=CHR$(227) 'Joueur 1
1260 j2$=CHR$(231) 'Joueur 2
```

Les erreurs de manipulation des joueurs sont signalées par des messages écrits avec une encre tantôt rouge tantôt noire :

```
1270 INK 2,0,6 'Encre pour signaler une erreur
```

Les cases sont mémorisées dans un tableau de 6 cases sur 6. Les lignes 1280 à 1350 définissent ce tableau et l'initialisent à 0.

Le sous-programme se termine par l'affichage de l'écran de jeu lignes 1420 à 1500. Cet affichage est simplifié à l'extrême, car les chaînes qui représentent les barres ont été calculées précédemment. Les chaînes **s2\$**, **s3\$** et **s4\$** sont affichées répétitivement à l'intérieur d'une boucle FOR NEXT :

```
1450 FOR i=1 TO 6
1460 LOCATE 13,1+i*3 : PRINT s2$
...
1490 NEXT i
```

Le second sous-programme est le plus important du programme : il régit le fonctionnement de chaque partie. La ligne 1560 change la main :

```
1560 IF j=1 THEN j$=j2$ : j=2 ELSE j$=j1$ : j=1
```

Les lignes 1570 à 1590 signalent quel joueur a la main et demandent au joueur d'entrer une commande.

La commande entrée est mise en majuscule ligne 1600 et analysée entre les lignes 1620 et 1640. Le sous-programme de déplacement ou de placement est exécuté en fonction de la commande entrée :

```
1620 IF MID$(jj$,1,1)="G" THEN GOSUB 1710
```

...

```
1640 IF MID$(jj$,1,1)="L" AND MID$(jj$,3,1)="C" THEN GOSUB 2070
```

Ces sous-programmes stockent la valeur 1 dans la variable **ok** lorsque la commande entrée est valide.

Lorsque la variable **ok** vaut 1, les tableaux **T** (tableau de jeu) et **D** (décalage des barres) sont mis à jour et le nouvel écran de jeu est affiché.

Lorsque la variable **ok** vaut 0, un message d'erreur est affiché sur l'écran :

```
1650 IF ok=1 THEN GOSUB 2540 ELSE GOSUB 2140
```

La ligne 1670 teste si trois pions de même couleur sont alignés.

La ligne 1680 redonne le contrôle à la ligne 1520 (boucle de jeu) si aucun des joueurs n'a aligné trois pions.

Le programme se poursuit par les sous-programmes de déplacement, de placement, d'affichage de l'écran de jeu, et de test de fin de jeu.

Analysons le fonctionnement du sous-programme de déplacement vers la gauche. Le déplacement vers la droite est tout à fait similaire.

Ce sous-programme se trouve entre les lignes 1710 et 1870. La première action effectuée consiste à tester la validité de la commande. La commande sera considérée valide si le numéro de la barre à déplacer est compris entre 1 et 6 :

```
1750 I=VAL(MID$(jj$,2,1))
```

```
1760 IF I<7 AND I>0 THEN 1850
```

Dans le cas contraire, un message d'erreur est affiché sur l'écran :

```
1770 PEN 2
```

```
1780 LOCATE 1,23
```

```
1790 PRINT"Commande invalide"
```

Une temporisation affiche le message d'erreur pendant un court instant. Le message est alors effacé :

```
1800 FOR i=1 TO 1000:NEXT i
1810 LOCATE 1,23
1820 PRINT SPACE$(20)
1830 PEN 1
```

Lorsque la commande a été acceptée, le tableau des décalages est mis à jour :

```
1850 d(l)=d(l)-1
```

et la variable `ok` est mise à 1 pour éviter l'affichage d'un message d'erreur :

```
1860 ok=1
```

Le sous-programme de placement d'un pion se trouve entre les lignes 2070 et 2240.

La ligne et la colonne du placement sont extraites de la commande lignes 2110 et 2120 :

```
2110 l=VAL(MID$(jj$,2,1)) 'Ligne du placement
2120 c=VAL(MID$(jj$,4,1)) 'Colonne du placement
```

Si la ligne et la colonne ont une valeur correcte (c'est-à-dire comprise entre 1 et 6), et si la case pointée est vide, le placement est effectif. Dans le cas contraire, un message d'erreur est affiché brièvement sur l'écran.

```
2130 IF (l>0) AND (l<7) AND (c>0) AND (c<7) AND (t(l,c)=0) THEN
2220
```

L'affichage du message d'erreur se fait entre les lignes 2140 et 2210. Il est similaire à celui dont nous avons déjà parlé. Nous n'y reviendrons pas.

Lorsque la commande est correcte, le programme initialise le tableau `T` en conséquence, et met la variable `ok` à 1 :

```
2220 t(l,c)=j
2230 ok=1
```

Le sous-programme de test de fin de jeu se trouve entre les lignes 2260 et 2530.

Ce test se fera dans le tableau `u` qui tiendra compte des décalages de chaque barre.

Dans un premier temps, le tableau `u` est mis à zéro à l'aide de deux boucles imbriquées lignes 2300 à 2340.

Dans un second temps, le tableau **t** est recopié dans le tableau **u** en tenant compte des décalages (tableau **d**).

Enfin dans un troisième temps, le programme teste si trois pions de même couleur se trouvent disposés côte à côte. Ce test se fait à l'aide de deux blocs de deux boucles imbriquées lignes 2400 à 2510.

La variable **g** est initialisée à 1 si trois pions du joueur 1 se trouvent côte à côte et à 2 si trois pions du joueur 2 se trouvent côte à côte.

Le sous-programme d'affichage de l'écran de jeu pendant la partie se trouve entre les lignes 2540 et 2730. Sa première partie (lignes 2580 à 2650) est la copie conforme des instructions d'affichage déjà étudiées dans le sous-programme d'initialisation.

Les instructions suivantes affichent les pions dans les barres :

```
2660 FOR i=1 TO 6
2670   FOR k=1 TO 6
2680     IF t(i,k)=0 THEN 2710
2690     LOCATE 13+d(i)*2+k*2-1,2+i*3
2700     IF t(i,k)=1 THEN PRINT j1$ ELSE PRINT j2$
2710   NEXT k
2720 NEXT i
```

Enfin, le programme se termine par le sous-programme de félicitations (lignes 2750 à 2840). Ce sous-programme affiche un message de félicitations qui tient compte du vainqueur.