

## 9/7.6

# Le chercheur de diamants

---

Un chercheur de diamants se promène dans une carrière parsemée de mines. Heureusement, il possède un appareil capable de le renseigner sur la proximité et sur le nombre de mines autour de sa position actuelle. Le but du jeu est de découvrir les dix diamants sans mettre le pied sur une mine. Ce jeu d'apparence assez simple vous passionnera sans doute si vous aimez les jeux de logique et de réflexion.

Le jeu est proposé en version Basic et Turbo Pascal.

**COMMENT UTILISER LE PROGRAMME**

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

```

1000 ' =====
1010 ' Le chercheur de diamants
1020 ' =====
1030 '
1040 GOSUB 1080 ' Initialisation
1050 GOSUB 2030 ' Jeu
1060 GOSUB 2660 ' Fin du jeu
1070 END
1080 '
1090 '-----
1100 ' Initialisation
1110 '-----
1120 '
1130 ' - - - - -
1140 ' Caracteres graphiques
1150 ' - - - - -
1160 '
1170 c1$=CHR$(150)
1180 c2$=CHR$(154)
1190 c3$=CHR$(158)
1200 c4$=CHR$(156)
1210 c5$=CHR$(149)
1220 c6$=CHR$(159)
1230 c7$=CHR$(147)
1240 c8$=CHR$(155)
1250 c9$=CHR$(153)
1260 c10$=CHR$(151)
1270 c11$=CHR$(157)
1280 c12$=CHR$(207)
1290 c13$=CHR$(241)
1300 c14$=CHR$(243)
1310 s1$=c3$+c2$
1320 s2$=c1$+c2$+s1$+s1$+s1$+s1$+s1$+s1$+s1$+s1$+c4$
1330 s3$=c5$+c12$+c5$+c12$+c5$+c12$+c5$+c12$+c5$+c12$+c5$+c1
2$+c5$+c12$+c5$+c12$+c5$+c12$+c5$
1340 s4$=c10$+c2$+c6$+c2$+c6$+c2$+c6$+c2$+c6$+c2$+c6$+c2$+c6
$+c2$+c6$+c2$+c6$+c2$+c11$
1350 s5$=c7$+c2$+c8$+c2$+c8$+c2$+c8$+c2$+c8$+c2$+c8$+c2$+c8$
+c2$+c8$+c2$+c8$+c2$+c9$
1360 '
1370 ' - - - - -
1380 ' Affichage du decor du jeu
1390 ' - - - - -
1400 '
1410 MODE 1
1420 LOCATE 7,1
1430 PRINT"Le chercheur de diamants"
1440 LOCATE 10,4
1450 PRINT s2$
1460 FOR i=1 TO 16 STEP 2

```

```

1470 LOCATE 10,i+4
1480 PRINT s3$
1490 LOCATE 10,i+5
1500 PRINT s4$
1510 NEXT i
1520 LOCATE 10,21
1530 PRINT s3$
1540 LOCATE 10,22
1550 PRINT s5$
1560 '
1570 ' - - - - -
1580 ' Pose des mines
1590 ' - - - - -
1600 '
1610 FOR i=1 TO 15
1620   x=INT(RND(1)*9)+1
1630   y=INT(RND(1)*9)+1
1640   IF (t(x,y)<>9) THEN t(x,y)=9 ELSE i=i-1
1650 NEXT i
1660 '
1670 ' - - - - -
1680 ' Calcul du nombre de mines a proximite
1690 ' - - - - -
1700 '
1710 FOR i=1 TO 9
1720   FOR j=1 TO 9
1730     nb=0
1740     IF (i<>1) AND (j<>1) AND (t(i,j)<>9) THEN IF (t(i-1
,j-1)=9) THEN nb=nb+1
1750     IF (i<>1) AND (t(i,j)<>9) THEN IF (t(i-1,j)=9) THEN
nb=nb+1
1760     IF (i<>1) AND (j<>9) AND (t(i,j)<>9) THEN IF (t(i-1
,j+1)=9) THEN nb=nb+1
1770     IF (j<>1) AND (t(i,j)<>9) THEN IF (t(i,j-1)=9) THEN
nb=nb+1
1780     IF (j<>9) AND (t(i,j)<>9) THEN IF (t(i,j+1)=9) THEN
nb=nb+1
1790     IF (i<>9) AND (j<>1) AND (t(i,j)<>9) THEN IF (t(i+1
,j-1)=9) THEN nb=nb+1
1800     IF (i<>9) AND (t(i,j)<>9) THEN IF (t(i+1,j)=9) THEN
nb=nb+1
1810     IF (i<>9) AND (j<>9) AND (t(i,j)<>9) THEN IF (t(i+1
,j+1)=9) THEN nb=nb+1
1820     IF (t(i,j)<>9) THEN t(i,j)=nb
1830   NEXT j
1840 NEXT i
1850 '
1860 ' - - - - -
1870 ' 10 diamants repartis aleatoirement
1880 ' - - - - -

```

```

1890 '
1900 FOR i=1 TO 10
1910   x=INT(RND(1)*9)+1
1920   y=INT(RND(1)*9)+1
1930   IF (t(x,y)=0) THEN t(x,y)=99 ELSE i=i-1
1940 NEXT i
1950 '
1960 ' - - - - -
1970 ' Affichage des fleches de selection
1980 ' - - - - -
1990 '
2000 LOCATE 11,3 : PRINT c13$ : x1=11 : y1=3
2010 LOCATE 9,5 : PRINT c14$ : x2=9 : y2=5
2020 RETURN
2030 '
2040 '-----
2050 ' Deroulement d'une partie
2060 '-----
2070 '
2080 a$=INKEY$
2090 IF a$="" THEN 2080
2100 a=ASC(a$)
2110 '
2120 ' - - - - -
2130 ' Deplacement vers la droite
2140 ' - - - - -
2150 '
2160 IF (a<>243) THEN 2270
2170 IF (x1=27) THEN 2540
2180 LOCATE x1,y1:PRINT" "
2190 x1=x1+2
2200 LOCATE x1,y1:PRINT c13$
2210 GOTO 2540
2220 '
2230 ' - - - - -
2240 ' Deplacement vers la gauche
2250 ' - - - - -
2260 '
2270 IF (a<>242) THEN 2380
2280 IF (x1=11) THEN 2540
2290 LOCATE x1,y1:PRINT" "
2300 x1=x1-2
2310 LOCATE x1,y1:PRINT c13$
2320 GOTO 2540
2330 '
2340 ' - - - - -
2350 ' Deplacement vers le bas
2360 ' - - - - -
2370 '
2380 IF (a<>241) THEN 2490

```

```
2390 IF (y2=21) THEN 2540
2400 LOCATE x2,y2:PRINT" "
2410 y2=y2+2
2420 LOCATE x2,y2:PRINT c14$
2430 GOTO 2540
2440 '
2450 ' - - - - -
2460 ' Deplacement vers le haut
2470 ' - - - - -
2480 '
2490 IF (a<>240) THEN 2540
2500 IF (y2=5) THEN 2540
2510 LOCATE x2,y2:PRINT" "
2520 y2=y2-2
2530 LOCATE x2,y2:PRINT c14$
2540 '
2550 ' - - - - -
2560 ' Affichage de la case
2570 ' - - - - -
2580 '
2590 IF UPPER$(a$)="Q" THEN RETURN
2600 IF a<>13 THEN 2030 'Touche interdite
2610 ca=t((y2-3)/2,(x1-9)/2)
2620 IF (ca<>9) AND (ca<>99) THEN LOCATE x1,y2:PRINT RIGHT$(
STR$(ca),1):GOTO 2030
2630 IF (ca=99) THEN LOCATE x1,y2:PRINT"*":d=d+1 'Diamant
2640 IF (ca=99) THEN IF (d=10) THEN RETURN ELSE 2030
2650 IF (ca=9) THEN LOCATE x1,y2:PRINT"B":RETURN 'Bombe
2660 '
2670 ' -----
2680 ' Fin de la partie
2690 ' -----
2700 '
2710 LOCATE 1,23
2720 IF UPPER$(a$)="Q" THEN PRINT"Arret avant la fin de la p
artie.":GOTO 2750
2730 IF d=10 THEN PRINT"Bravo. Vous voila riche."
2740 IF d<>10 THEN PRINT"Vous etes tombe sur une mine..."
2750 RETURN
```

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

```

Program Diamants;
{ ===== }
{ Le chercheur de diamants }
{ ===== }

VAR
  c1, c2, c3, c4,
  c5, c6, c7, c8,
  c9, c10, c11, c12,
  c13, c14      : Char;
  s1, s2, s3, s4, s5 : String[30];
  I, J : Byte;
  X, Y : Byte;
  X1, Y1, X2, Y2 : Byte;
  T : Array[1..9,1..9] of Byte;
  Nb : Byte;
  as : Char;
  a, CA : Byte;
  d : Byte;

Procedure Init;
{ ----- }
{ Initialisation du jeu }
{ ----- }
begin
  { Caracteres graphiques }
  c1:=CHR(150);
  c2:=CHR(154);
  c3:=CHR(158);
  c4:=CHR(156);
  c5:=CHR(149);
  c6:=CHR(159);
  c7:=CHR(147);
  c8:=CHR(155);
  c9:=CHR(153);
  c10:=CHR(151);
  c11:=CHR(157);
  c12:=CHR(207);
  c13:=CHR(241);
  c14:=CHR(243);
  s1:=c3+c2;
  s2:=Concat(c1,c2,s1,s1,s1,s1,s1,s1,s1,s1,c4);
  s3:=c5+c12+c5+c12+c5+c12+c5+c12+c5+c12+c5+c12+c5+c12+c5+c12+c5;
  s4:=c10+c2+c6+c2+c6+c2+c6+c2+c6+c2+c6+c2+c6+c2+c6+c2+c6+c2+c11;
  s5:=c7+c2+c8+c2+c8+c2+c8+c2+c8+c2+c8+c2+c8+c2+c8+c2+c8+c2+c9;

  { Affichage du decor du jeu }
  ClrScr;
  GotoXY(7,1);
  Write('Le chercheur de diamants');

```

```

GotoXY(10,4);
Write(s2);
I:=1;
Repeat
  GotoXY(10,i+4);
  Write(s3);
  GotoXY(10,i+5);
  Write(s4);
  I:=I+2;
until (I=17);
GotoXY(10,21);
Write(s3);
GotoXY(10,22);
Write(s5);

{ Pose des mines }
Randomize;
For i:=1 to 9 do
  For J:=1 to 9 do
    T[I,J]:=0;
FOR i:=1 TO 15 do
begin
  x:=Round(Random(9))+1;
  y:=Round(Random(9))+1;
  IF (t[x,y]<>9)
  THEN t[x,y]:=9
  ELSE i:=i-1;
end;

{ Calcul du nombre de mines a proximite }
FOR i:=1 TO 9 do
  FOR j:=1 TO 9 do
  begin
    nb:=0;
    IF (i<>1) AND (j<>1) AND (t[i,j]<>9)
    THEN IF (t[i-1,j-1]=9)
      THEN nb:=nb+1;
    IF (i<>1) AND (t[i,j]<>9)
    THEN IF (t[i-1,j]=9)
      THEN nb:=nb+1;
    IF (i<>1) AND (j<>9) AND (t[i,j]<>9)
    THEN IF (t[i-1,j+1]=9)
      THEN nb:=nb+1;
    IF (j<>1) AND (t[i,j]<>9)
    THEN IF (t[i,j-1]=9)
      THEN nb:=nb+1;
    IF (j<>9) AND (t[i,j]<>9)
    THEN IF (t[i,j+1]=9)
      THEN nb:=nb+1;
    IF (i<>9) AND (j<>1) AND (t[i,j]<>9)

```

```

    THEN IF (t[i+1,j-1]=9)
        THEN nb:=nb+1;
    IF (i<>9) AND (t[i,j]<>9)
    THEN IF (t[i+1,j]=9)
        THEN nb:=nb+1;
    IF (i<>9) AND (j<>9) AND (t[i,j]<>9)
    THEN IF (t[i+1,j+1]=9)
        THEN nb:=nb+1;
    IF (t[i,j]<>9) THEN t[i,j]:=nb;
end;

{ 10 diamants repartis aleatoirement }
I:=1;
Repeat
    x:=Random(9)+1;
    y:=Random(9)+1;
    IF (t[x,y]=0) THEN
    begin
        t[x,y]:=99;
        I:=I+1;
    end;
until (I=11);

{ Affichage des fleches de selection }
GotoXY(11,3);
Write(c13);
x1:=11;
y1:=3;
GotoXY(9,5);
Write(c14);
x2:=9;
y2:=5;
d:=0;
end;

Procedure Jeu;
{ ----- }
{ Deroulement d'une partie }
{ ----- }
begin
    Repeat
        Repeat
            Until KeyPressed;
        Read(Kbd,as);
        a:=Ord(as);

        Case a of
            { Deplacement vers la droite }
            243 : IF (x1<>27) THEN
                begin

```

```
        GotoXY(x1,y1);
        Write(' ');
        x1:=x1+2;
        GotoXY(x1,y1);
        Write(c13);
    end;

    { Deplacement vers la gauche }
    242 : IF (x1<>11) THEN
        begin
            GotoXY(x1,y1);
            Write(' ');
            x1:=x1-2;
            GotoXY(x1,y1);
            Write(c13);
        end;

    { Deplacement vers le bas }
    241 : IF (y2<>21) THEN
        begin
            GotoXY(x2,y2);
            Write(' ');
            y2:=y2+2;
            GotoXY(x2,y2);
            Write(c14);
        end;

    { Deplacement vers le haut }
    240 : IF (y2<>5) THEN
        begin
            GotoXY(x2,y2);
            Write(' ');
            y2:=y2-2;
            GotoXY(x2,y2);
            Write(c14);
        end;

end;
{ Affichage de la case }
IF (a=13) THEN
begin
    ca:=t[Round((y2-3)/2),Round((x1-9)/2)];
    IF (ca<>9) AND (ca<>99) THEN
    begin
        GotoXY(x1,y2);
        Write(ca);
    end;
    IF (ca=99) THEN
    begin
        GotoXY(x1,y2);
```

```

        Write('*');
        d:=d+1; { 1 diamant de plus }
    end;
    IF (ca=9) THEN
    begin
        GotoXY(x1,y2);
        Write('B');
    end;
end;
Until (D=10) or (ca=9) or (UpCase(a)='Q');
end;

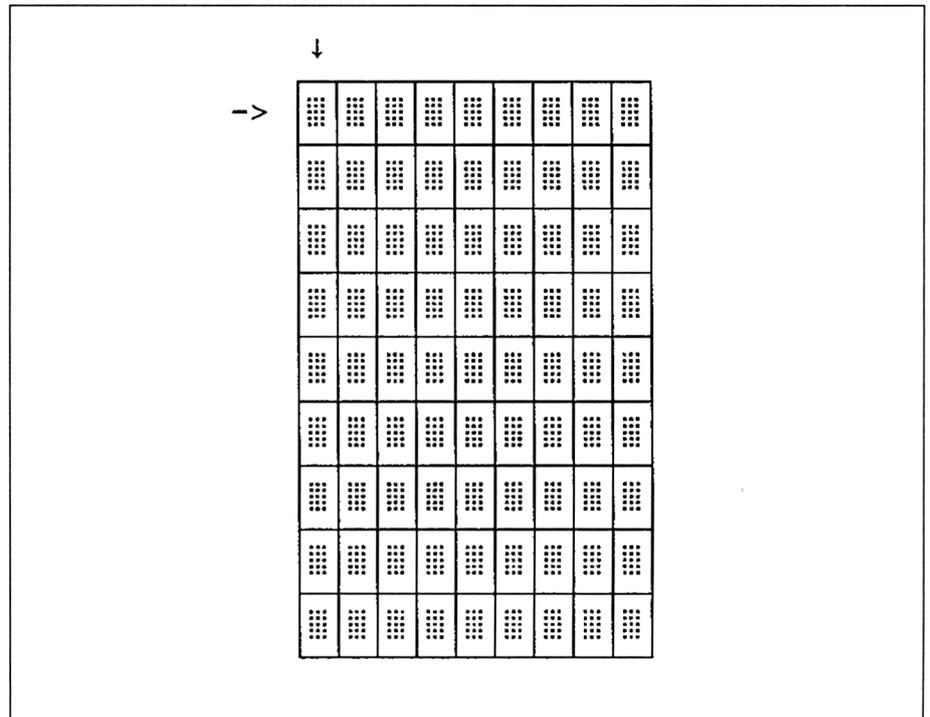
Procedure Fin;
{ ----- }
{ Fin de la partie }
{ ----- }
begin
    GotoXY(1,23);
    IF (d=10)
    THEN Write('Bravo. Vous voila riche.')
    ELSE IF (UpCase(a)='Q') THEN Write('Arret avant la fin de la partie.')
        ELSE Write('Vous etes tombe sur une mine...');
end;

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

begin
    Init; { Initialisation du jeu      }
    Jeu;  { Deroulement d'une partie }
    Fin;  { Commentaire sur la partie }
end.

```

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



Pour jouer, déplacez les flèches de telle sorte que leur intersection se trouve sur la case de votre choix, et appuyez sur la touche **ENTER**.

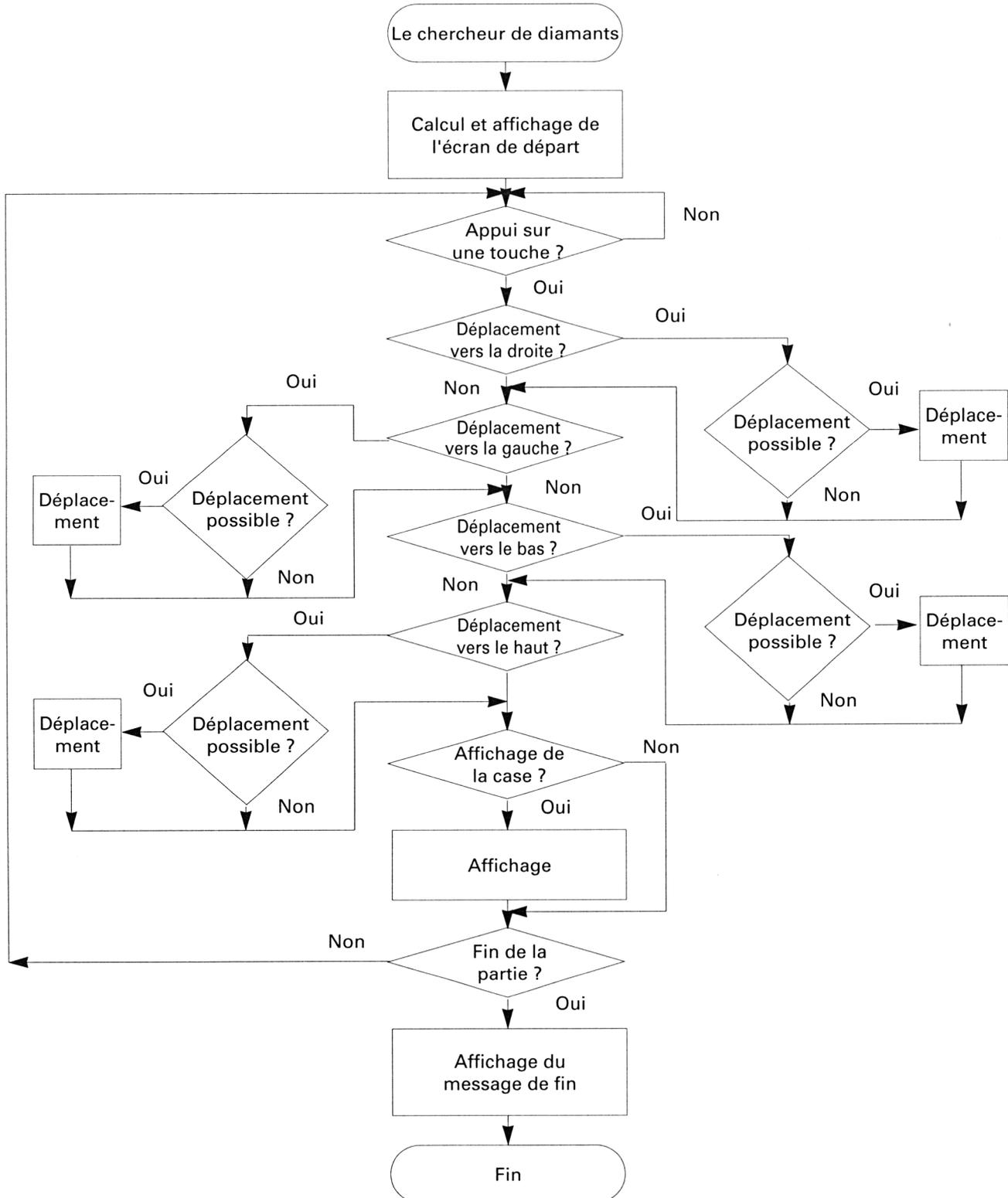
Trois choses peuvent se passer :

- la case ainsi désignée contient un diamant. Le diamant apparaît sur l'écran sous la forme d'une étoile ;
- la case désignée contient une mine. Un **"B"** est affiché sur l'écran, et la partie prend fin ;
- la case pointée ne contient ni diamant ni mine. Dans ce cas, une valeur numérique est affichée sur l'écran. Elle représente le nombre de mines situées sur la périphérie de la case. La valeur affichée est donc comprise entre 0 et 8.

Le jeu peut être quitté à tout instant en appuyant sur la touche **"Q"** du clavier.

## LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Etudions la version Turbo Pascal du programme.

Le programme met en œuvre trois procédures :

- **Initialisation** pour afficher l'écran de départ et initialiser les variables utilisées par le programme ;
- **Jeu** pour gérer les commandes entrées au clavier ;
- **Fin** pour afficher un message de fin de partie.

*Procédure Init :*

Le tableau de jeu est constitué de 9 cases sur 9. Ces cases sont affichées sur l'écran à l'aide de caractères graphiques. Pour faciliter leur manipulation, les caractères utilisés sont stockés dans des variables caractères :

```
c1 :=CHR(150);
```

```
...
```

```
c14 :=CHR(243);
```

Le tableau de jeu peut être créé à partir de quatre lignes : la première, la seconde, la troisième et la dernière. Ces chaînes sont calculées et stockées dans les variables **s2** à **s5**. Pour afficher le tableau de jeu, il suffit alors d'afficher répétitivement les chaînes appropriées :

```
GotoXY(10,4);
Write(s2);
I :=1;
Repeat
  GotoXY(10,i+4);
  Write(s3);
  GotoXY(10,i+5);
  Write(s4);
  I :=I+2;
until (I=17);
GotoXY(10,21);
Write(s3);
GotoXY(10,22);
Write(s5);
```

Le bloc d'instructions suivant initialise le tableau de jeu et pose quinze mines sur la surface de jeu. Les mines sont posées aléatoirement sur des cases inoccupées :

```
For i :=1 to 15 do
begin
  x :=Round(Random(9))+1;
  y :=Round(Random(9))+1;
  IF (t[x,y]< >9)
  THEN t[x,y] :=9
  ELSE i :=i-1;
end;
```

Le bloc d'instructions suivant stocke des informations numériques dans les cases qui entourent chaque mine. Pour ce faire, les huit cases qui entourent chaque case sont testées. Chaque mine détectée incrémente la variable **nb**. En fin de test, la valeur de la variable **nb** est stockée dans la case courante, si cette case est libre.

```

For i :=1 to 9 do
  For j :=1 to 9 do
    begin
      nb :=0
      IF (i< >1) AND (j< >1) AND (t[i,j]< > 9)
      THEN IF (t[i-1,j-1]=9)
        THEN nb :=nb+1
      ...
      IF (t[i,j]< >9) THEN t[i,j] :=nb;
    end
  
```

Le bloc suivant répartit aléatoirement dix diamants sur les cases libres.

Enfin, le dernier bloc de la procédure affiche les flèches de sélection "vers le bas" et "vers la droite".

*Procédure Jeu :*

La procédure Jeu est le cœur du programme. C'est elle qui gère les actions du joueur (déplacement des flèches, appui sur la touche **ENTER** ou sur la touche **Q**).

Les actions du clavier sont acquises à l'intérieur d'une boucle **Repeat Until** qui prend fin lorsque :

- le nombre de diamants découverts est égal à 10 ;
- le joueur tombe sur une mine ;
- la touche "Q" a été pressée.

```

Repeat
  ...
Until (D=10) or (ca=9) or (UpCase(a)='Q');
  
```

L'acquisition des touches actionnées se fait à l'aide de la fonction **Read** :

```

Repeat
Until KeyPresse;
Read (Kbd,as);
a :=Ord(as);
  
```

Le code de la touche pressée est détecté dans une structure **Case of**.

Les traitements liés aux touches de déplacement testent si les déplacements demandés sont possibles. Par exemple, un déplacement vers

la droite ne peut s'effectuer que si le repère ne se trouve pas à l'extrême droite du tableau :

```
243 : IF (x1 < > 27) THEN
```

Dans ce cas, l'ancien repère est effacé :

```
GotoXY(x1,y1);
Write(' ');
```

l'abscisse est incrémentée :

```
x1 :=x1+2;
```

et le nouveau repère est affiché :

```
GotoXY(x1,y1);
Write(c13);
```

Les autres déplacements s'effectuent selon le même principe. Nous n'y reviendrons pas.

Lorsque la touche ENTER est pressée :

```
IF (a=13) THEN
```

le programme teste la valeur de la case correspondante dans le tableau T :

```
ca :=t[Round((y2-3)/2),Round((x1-9)/2)];
```

Si la case pointée ne contient ni un diamant ni une mine, la valeur numérique correspondante est affichée :

```
IF (ca < > 9) AND (ca < > 99) THEN
begin
  GotoXY(x1,y2);
  Write(ca);
end;
```

Si la case pointée contient un diamant, un signe "\*" est affiché, et le nombre de diamants découverts est incrémenté d'un :

```
IF (ca=99) THEN
begin
  GotoXY(x1,y2);
  Write('*');
  d :=d+1;
end ;
```

Enfin, si la case pointée contient une mine, la lettre "B" est affichée :

```
IF (ca=9) THEN
begin
  GotoXY(x1,y2);
  Write('B');
end;
```

Si le nombre de diamants découverts est égal à 10 (**D=10**), si la case pointée contient une mine, ou encore si la touche "**Q**" est pressée, la procédure **Jeu** prend fin :

**Until (D=10) or (ca=9) or (UpCase(a)='Q');**

*Procédure Fin :*

La procédure **Fin** affiche un message de fin de partie qui dépend des conditions dans lesquelles s'est finie la partie :

- "**Bravo. Vous voilà riche.**" si les dix dimants ont été découverts ;
- "**Vous êtes tombé sur une mine...**" si la case pointée contenait une mine ;
- "**Arrêt avant la fin de la partie.**" si la touche "**Q**" a été pressée.

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