

9/7.4

Passage non protégé

Ce jeu d'arcades est présenté en version Basic et Turbo Pascal dans le but de vous permettre de mieux appréhender les similitudes entre ces deux langages.

Un piéton courageux doit traverser le plus souvent possible une autoroute à quatre voies sur laquelle les chauffards ne manquent pas. Il peut reculer ou avancer comme bon lui semble. La durée d'une partie est limitée dans le temps à une minute.

COMMENT UTILISER LE PROGRAMME

Si vous désirez utiliser la version Basic du langage, entrez le listing suivant :

```

1000 REM =====
1010 REM Passage non protege
1020 REM =====
1030 REM
1040 GOSUB 2000 'Presentation du jeu
1050 GOSUB 3000 'Redefinition des caracteres
1060 GOSUB 4000 'Initialisation des variables
1070 GOSUB 5000 'Affichage des elements du jeu
1080 GOSUB 6000 'Deroulement d'une partie
1090 GOSUB 7000 'Fin du jeu
1100 END
2000 REM -----
2010 REM Presentation du jeu
2020 REM -----
2030 REM
2040 MODE 1
2050 PRINT "           Passage non protege"
2060 PRINT "           -----"
2070 PRINT
2080 PRINT"Le but du jeu est de traverser la route"
2090 PRINT"le plus souvent possible en evitant les"
2100 PRINT"vehicules en circulation."
2110 PRINT"Les touches-fleches vers le haut et vers";
2120 PRINT"le bas permettent de deplacer le pieton."
2130 PRINT"Le temps est limite a une minute."
2140 PRINT
2150 PRINT"Appuyez sur une touche ..."
2160 a$=INKEY$
2170 IF a$="" THEN 2160
2180 RETURN
3000 REM -----
3010 REM Redefinition des caracteres
3020 REM -----
3030 REM
3040 SYMBOL AFTER 129
3050 FOR i=1 TO 2
3060   FOR j=1 TO 8
3070     READ a(j)
3080   NEXT j
3090   SYMBOL 128+i, a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8)
3100 NEXT i
3110 '
3120 DATA 0,28,28,8,28,8,8,20
3130 DATA 0,0,12,63,63,18,0,0
3140 RETURN
4000 REM -----
4010 REM Initialisation des variables
4020 REM -----
4030 REM
4040 vo=1 'Nombre de voies
4050 t(1)=9 : t(2)=9 'Position des voitures voies 1 et 2
4060 t(3)=1 : t(4)=1 'Position des voitures voies 3 et 4
4070 pj=0 'Position du joueur
4080 RETURN

```

```

5000 REM -----
5010 REM Affichage des elements du jeu
5020 REM -----
5030 REM
5040 MODE 1
5050 LOCATE 16,1
5060 PRINT "Score : 0"
5070 FOR i=1 TO 5
5080   FOR j=1 TO 19
5090     LOCATE 10+j,9+2*i
5100     PRINT "-"
5110   NEXT j
5120 NEXT i
5130 FOR i=1 TO 19
5140   LOCATE 10+i,15
5150   PRINT "="
5160 NEXT i
5170 LOCATE 28,12 : PRINT CHR$(130) 'Voiture 1
5180 LOCATE 28,14 : PRINT CHR$(130) 'Voiture 2
5190 LOCATE 12,16 : PRINT CHR$(130) 'Voiture 3
5200 LOCATE 12,18 : PRINT CHR$(130) 'Voiture 4
5210 LOCATE 20,20 : PRINT CHR$(29) 'Pieton
5220 RETURN
6000 REM -----
6010 REM Deroulement d'une partie
6020 REM -----
6030 REM
6040 tstart=TIME/300 'Comteur au debut de la partie
6050 a$=INKEY$
6060 IF a$ <> "" THEN GOSUB 8000 'Action du joueur
6070 GOSUB 9000 'Deplacement des voitures
6080 IF fin=0 THEN 6050
6090 RETURN
7000 REM -----
7010 REM Fin du jeu
7020 REM -----
7030 REM
7040 MODE 2
7050 IF fin>=5 THEN PRINT "Vous avez perri sous une voiture."
";GOTO 7080
7070 IF fin<4 THEN PRINT "Votre desir de passer vous a jete
sous une voiture."
7080 PRINT "Temps ecoule : ";INT((TIME/300-tstart));" seconde(
s). "
7090 PRINT "Score :";x;"passage(s). "
7100 RETURN
8000 REM -----
8010 REM Action du joueur
8020 REM -----
8030 REM
8040 IF ASC(a$)<>240 THEN 8280
8050 '
8060 ' Deplacement vers le haut
8070 '

```

```

8080 pj=pj+1
8090 FOR i=1 TO 4
8100   IF pj=5-i AND t(i)=5 THEN fin=2 'Collision avec une v
      oiture
8110 NEXT i
8120 IF pj<>5 THEN 8230
8130 LOCATE 20,12
8140 PRINT " "
8150 pj=0 : sc=sc+1
8160 LOCATE 20,20
8170 PRINT CHR$(129)
8180 PRINT CHR$(7);
8190 x=x+1
8200 LOCATE 23,1
8210 PRINT x
8220 RETURN
8230 LOCATE 20,20-2*pj
8240 PRINT CHR$(129)
8250 LOCATE 20,22-pj*2
8260 PRINT " "
8270 RETURN
8280 '
8290 ' Deplacement vers le bas
8300 '
8310 pj=pj-1
8320 FOR i=1 TO 4
8330 IF pj=5-i AND t(i)=5 THEN fin=3 'Collision avec une voi
      ture
8340 NEXT i
8350 IF pj=0 THEN pj=1:RETURN
8360 IF pj=-1 THEN pj=0
8370 LOCATE 20,20-2*pj
8380 PRINT CHR$(129)
8390 LOCATE 20,18-2*pj
8400 PRINT " "
8410 RETURN
9000 REM -----
9010 REM Deplacement des voitures
9020 REM -----
9030 REM
9040 IF (TIME/300)-tstart > 60 THEN fin=4 'Fin de la partie
9050 IF vo=1 OR vo=2 THEN 9250 'deplacement vers la gauche
9060 '-----
9070 ' Deplacement vers la droite
9080 '-----
9090 IF t(vo)<>9 THEN 9170
9100 LOCATE 28,10+2*vo
9110 PRINT " "
9120 LOCATE 12,10+2*vc
9130 PRINT CHR$(130)
9140 t(vo)=1
9150 vo=INT(RND(1)*4)+1
9160 RETURN
9170 IF t(vo)=4 AND pj=5-vo THEN fin=5 'Collision avec une v

```

```
oiture
9180 LOCATE 10+t(vo)*2,10+vo*2
9190 PRINT " "
9200 t(vo)=t(vo)+1
9210 LOCATE 10+t(vo)*2,10+vo*2
9220 PRINT CHR$(130)
9230 vo=INT(RND(1)*4)+1
9240 RETURN
9250 '- - - - -
9260 ' Deplacement vers la gauche
9270 '- - - - -
9280 IF t(vo)<>1 THEN 9360
9290 LOCATE 12,10+vo*2
9300 PRINT " "
9310 t(vo)=9
9320 LOCATE 28,10+vo*2
9330 PRINT CHR$(130)
9340 vo=INT(RND(1)*4)+1
9350 RETURN
9360 IF t(vo)=6 AND pj=5-vo THEN fin=6 'Collision avec une v
oiture
9370 LOCATE 10+t(vo)*2,10+vo*2
9380 PRINT " "
9390 t(vo)=t(vo)-1
9400 LOCATE 10+t(vo)*2,10+vo*2
9410 PRINT CHR$(130)
9420 vo=INT(RND(1)*4)+1
9430 RETURN
```

Si vous désirez utiliser la version Turbo Pascal, entrez le listing suivant :

```

Program Passage_Non_Protege;
{=====}
{ Traversee dangereuse d'un piéton sur }
{ une autoroute a quatre voies.      }
{=====}

Var VO      : Byte; { Voie }
    Pj      : Byte; { Position du joueur }
    T       : Array[1..4] of Byte; { Position des voitures }
    I,J     : Byte; { Index de boucles }
    K       : Integer; { Index de boucle }
    Ch      : Char; { Caractere lu au clavier }
    Fin     : Byte; { Indicateur de fin de partie }
    SC      : Byte; { Nombre de traversees }
    X       : Byte; { Score }
    Temps   : Integer; { Compteur de passage }

Procedure Presentation;
{-----}
{ Presentation du jeu }
{-----}

begin
  ClrScr;
  Writeln('          Passage non protege');
  Writeln('          -----');
  Writeln;
  Writeln('Le but du jeu est de traverser la route');
  Writeln('le plus souvent possible en evitant les');
  Writeln('vehicules en circulation. ');
  Writeln('Les touches--fleches vers le haut et vers');
  Writeln('le bas permettent de deplacer le piéton. ');
  Writeln('Le temps est limite a une minute. ');
  Writeln;
  Write('Appuyez sur une touche ... ');
  Repeat
  until KeyPressed;
end;

Procedure Jeu;
{-----}
{ Deroulement du jeu }
{-----}

begin
  ClrScr;
  { Initialisation des variables }
  VO:=1;           { Nombre de voies      }
  T[1]:=9; T[2]:=9; { Voies 1 et 2      }
  T[3]:=1; T[4]:=1; { Voies 3 et 4      }

```

```

Pj:=0;           ( Position joueur      )
Fin:=0;         ( La partie commence  )
SC:=0;         ( Aucune traversée     )
Temps:=0;      ( Init compteur de jeu )
X:=0;          ( Nombre de passages  )
Randomize;     ( Init gene aleatoire  )

( - - - - - )
( Affichage du jeu )
( - - - - - )
GotoXY(16,1);
Writeln('Score : 0');
For I:=1 to 5 do
  For J:=1 to 19 do
    begin
      GotoXY(10+J,9+2*I);
      Write('-');
    end;
  For I:=1 to 19 do
    begin
      GotoXY(10+I,15);
      Write('=');
    end;
  GotoXY(28,12); Write(Chr(242)); ( Voiture 1 )
  GotoXY(28,14); Write(Chr(242)); ( Voiture 2 )
  GotoXY(12,16); Write(Chr(243)); ( Voiture 3 )
  GotoXY(12,18); Write(Chr(243)); ( Voiture 4 )
  GotoXY(20,20); Write(Chr(249)); ( Piéton )

Repeat
  For K:=1 to 2000 do; ( Ralentisseur )

  If KeyPressed then
  begin
    Read(Kbd,Ch);
    ( - - - - - )
    ( Action du joueur )
    ( - - - - - )
    Case Ord(Ch) of
      240 :
      begin
        ( - - - - - )
        ( Deplacement vers le haut )
        ( - - - - - )
        Pj:=Pj+1;
        For I:=1 to 4 do
          If (Pj=5-I) and (T[I]=5) then Fin:=2; ( Collision
)

        If Pj=5 then
        begin
          GotoXY(20,12);
          Write(' ');
          Pj:=0;
          SC:=SC+1;

```

```

        GotoXY(20,20);
        Write(Chr(249));
        Write(Chr(7));
        X:=X+1;
        GotoXY(23,1);
        Write(X);
    en'
    else
    begin
        GotoXY(20,20-2*pj);
        Write(Chr(249));
        GotoXY(20,22-Pj*2);
        Write(' ');
    end;
end;
241 :
begin
    { - - - - - }
    { Deplacement vers le bas }
    { - - - - - }
    Pj:=Pj-1;
    For I:=1 to 4 do
        If (Pj=5-I) and (T[I]=5) then Fin:=3; { Collision
}
        If Pj=0
        then
            Pj:=1
        else
        begin
            If Pj=-1 then Pj:=0;
            GotoXY(20,20-2*Pj);
            Write(Chr(249));
            GotoXY(20,18-2*Pj);
            Write(' ');
        end;
    end;
end;
end;
}
{ - - - - - }
{ Deplacement des voitures }
{ - - - - - }
Temps:=Temps+1;
If Temps=300
then Fin:=1; { Fin du temps imparti }
If (VO=1) or (VO=2) then
begin
    { - - - - - }
    { Deplacement vers la gauche }
    { - - - - - }
    If T[VO]=1 then
    begin
        GotoXY(12,10+VO*2);
        Write(' ');
    end;
end;

```

```

    T[V0]:=9;
    GotoXY(28,10+V0*2);
    Write(Chr(242));
    V0:=Round(Random*3)+1;
end
else
begin
    If (T[V0]=6) and (Pj=5-V0) then
        Fin:=6; { Collision }
    GotoXY(10+T[V0]*2,10+V0*2);
    Write(' ');
    T[V0]:=T[V0]-1;
    GotoXY(10+T[V0]*2,10+V0*2);
    Write(Chr(242));
    V0:=Round(Random*3)+1;
end;
end
else
begin
    {-----}
    { Deplacement vers la droite }
    {-----}
    If T[V0]=9 then
    begin
        GotoXY(28,10+2*V0);
        Write(' ');
        GotoXY(12,10+2*V0);
        Write(Chr(243));
        T[V0]:=1;
        V0:=Round(Random*3)+1;
    end
    else
    begin
        If (T[V0]=4) and (Pj=5-V0)
            then Fin:=5; { Collision }
        GotoXY(10+T[V0]*2,10+V0*2);
        Write(' ');
        T[V0]:=T[V0]+1;
        GotoXY(10+T[V0]*2,10+V0*2);
        Write(Chr(243)),
        V0:=Round(Random*3)+1;
    end;
end;
until (Fin <> 0);
end;

```

```

Procedure Fin_Jeu;
{-----}
{ Commentaire de fin de partie }
{-----}

begin

```

```

    ClrScr;
    If Fin>=5
    then Writeln('Vous avez perri sous une voiture. ');
    If (Fin<=4) and (Fin<>1)
    then Writeln('Votre desir de passer vous a jete sous un
e voiture. ');
    If Fin=1
    then Writeln('Le temps imparti est ecoule. ');
    Writeln('Score : ',X,' passage(s). ');
end;

{=====}
{ PROGRAMME PRINCIPAL }
{=====}

begin
  Presentation; { Presentation du jeu      }
  Jeu;          { Deroulement d'une partie }
  Fin_Jeu;     { Commentaire de fin de partie }
end.

```

Exécutez le programme. La règle du jeu est affichée sur l'écran. Appuyez sur une touche quelconque du clavier pour commencer à jouer.

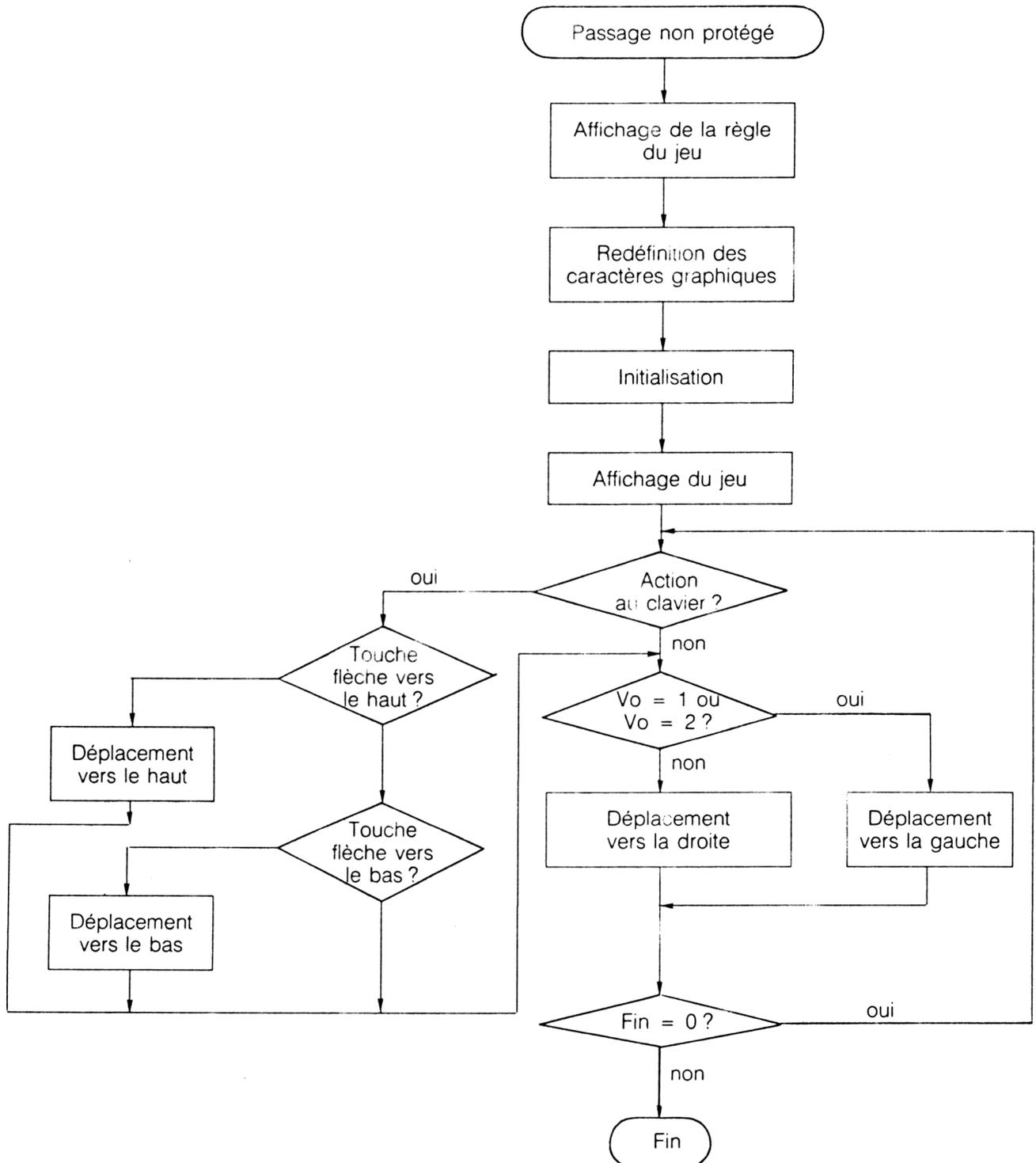
Le tracé de l'autoroute apparaît sur l'écran. Pour faire avancer ou reculer le piéton, utiliser (respectivement) les touches flèches vers le haut et vers le bas.

Chaque traversée est signalée par un bip sonore. Le nombre de traversées est affiché en permanence en haut de l'écran.

Lorsque le temps imparté est écoulé ou lorsque le piéton se fait renverser par une voiture, un commentaire est affiché sur l'écran.

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Programme en version Basic

Le programme principal fait appel à six sous-programmes :

```
1040 GOSUB 2000 'Presentation du jeu
1050 GOSUB 3000 'Redefinition des caracteres
1060 GOSUB 4000 'Initialisation des variables
1070 GOSUB 5000 'Affichage des elements du jeu
1080 GOSUB 6000 'Deroulement d'une partie
1090 GOSUB 7000 'Fin du jeu
```

La présentation du jeu est élémentaire. L'écran est tout d'abord effacé et mis en mode 1 :

```
2040 MODE 1
```

La règle du jeu est ensuite affichée à l'aide d'instructions PRINT :

```
2080 PRINT "Le but du jeu est de traverser la route"
...
2130 PRINT "Le temps est limite a une minute."
```

Le programme demande à l'utilisateur d'appuyer sur une touche du clavier pour commencer à jouer. L'attente de l'appui sur une touche se fait en lignes 2160 et 2170 à l'aide d'une boucle sur l'instruction INKEY\$ qui prend fin lorsque INKEY\$ est non vide, c'est-à-dire lorsque le joueur a appuyé sur une touche :

```
2160 a$ = INKEY$
2170 IF a$ = "" THEN 2160
```

Le contrôle est ensuite rendu au programme principal qui active le sous-programme de redéfinition des caractères. Ce programme affecte aux caractères de codes ASCII 129 et 130 (à l'aide des instructions SYMBOL AFTER et SYMBOL) la définition du piéton et des voitures :

```
3040 SYMBOL AFTER 129
3050 FOR i=1 TO 2
3060   FOR j=1 TO 8
3070     READ a(j)
3080   NEXT j
3090   SYMBOL 128+i,a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8)
3100 NEXT i
```

Les variables et constantes du jeu sont ensuite initialisées à l'aide du sous-programme situé entre les lignes 4000 et 4080.

Les éléments du jeu (autoroute, voitures et piéton) sont ensuite affichés à l'aide du sous-programme situé entre les lignes 5000 et 5220.

Le sous-programme débutant en ligne 6000 est alors activé, et la partie commence.

L'heure de début de partie est stockée dans la variable **tstart** :

```
6040 tstart = TIME/300
```

Si une touche du clavier a été actionnée, le sous-programme débutant en 8000 chargé des déplacements du piéton est exécuté :

```
6050 a$ = INKEY$
```

```
6060 IF a$ < > " " THEN GOSUB 8000 'Action du joueur
```

Le sous-programme chargé du déplacement des voitures est ensuite systématiquement exécuté :

```
6070 GOSUB 9000 'Déplacement des voitures
```

Le jeu se poursuit tant que la variable **fin** est nulle. Dans le cas contraire, cela signifie que le temps imparti est écoulé ou que le piéton a été écrasé :

```
6080 if fin = 0 then 6050
```

Le sous-programme chargé des déplacements du piéton est relativement simple. Si la touche pressée est la touche-flèche vers le haut, le piéton se déplace vers le haut (lignes 8080 à 8270). Si la touche pressée est la touche-flèche vers le bas, le piéton se déplace vers le bas (lignes 8310 à 8400).

Lorsque le temps imparti est écoulé, la variable **fin** est initialisée à 4, ce qui aura pour effet d'arrêter le jeu lors du prochain passage dans la boucle principale :

```
9040 IF (TIME/300) - tstart > 60 then fin = 4
```

Dans le cas où la variable **vo** vaut 1 ou 2, la voiture sélectionnée doit se déplacer vers la gauche. Le programme donne le contrôle à la ligne 9250. Dans le cas contraire, le programme se poursuit en séquence.

Programme en version Turbo Pascal

Le programme débute par la déclaration des variables. Dans un souci d'économie de place, la plupart des variables entières sont de type Byte.

Le programme est structuré en trois procédures principales.

La procédure **Presentation** est comparable au sous-programme Basic situé entre les lignes 2000 et 2180. L'attente de l'appui sur une touche du clavier est réalisé à l'intérieur d'une boucle Repeat qui prend fin lorsque la fonction **KeyPressed** vaut True :

```
Repeat  
until KeyPressed ;
```

La procédure **Jeu** contient la plupart des actions vitales du programme. Les variables du jeu sont initialisées. Remarquez l'initialisation de certaines variables qui n'étaient pas initialisées dans la version Basic ; ceci car le Basic affecte automatiquement la valeur 0 à toutes les variables numériques lorsque le programme est exécuté, alors que le Turbo Pascal n'affecte aucune valeur particulière aux variables lors de l'exécution du programme.

Le décor est ensuite affiché à l'aide d'instructions **GotoXY** (équivalent de **LOCATE** en Basic) et **Write** (équivalent de **PRINT** en Basic).

Remarquez la présence d'un ralentisseur :

```
For K: = 1 to 2000 do; { Ralentisseur }
```

Enlevez-le, et vous constaterez que le Turbo Pascal s'exécute bien plus vite que le Basic...

Lorsque le joueur a appuyé sur une touche :

```
If KeyPressed then
```

La touche pressée est lue et stockée dans la variable Ch :

```
Read(Kbd,Ch);
```

Les instructions exécutées dépendent de la touche pressée. Cette touche est identifiée dans une structure **Case Of** :

```
Case Ord(Ch) Of
240:
begin
  { Déplacement vers le haut }
  ...
end;
241:
begin
  { Déplacement vers le bas }
  ...
end;
```

La variable **BASIC TIME** n'étant pas disponible en Turbo Pascal, le programme la simule à l'aide de la variable **Temps** qui est incrémentée d'un à chaque déplacement d'une voiture. Lorsque cette variable est égale à 300 (environ au bout d'une minute), la variable **Fin** est initialisée à 1 :

```
Temps: = Temps + 1;
If Temps = 300
then Fin: = 1; { Fin du temps imparti }
```

Le déplacement des voitures vers la droite ou vers la gauche suit la même logique que celle du programme Basic. Remarquez la différence d'expression relative au tirage aléatoire d'un nombre compris entre 1 et 4 :

En Turbo Pascal : $VO := \text{Round}(\text{Random} * 3) + 1$;
En Basic $vo = \text{INT}(\text{RND}(1) * 4) + 1$

La procédure **Fin_Jeu** est exécutée lorsque le programme détecte une valeur non nulle dans la variable **Fin**. Un message approprié est alors affiché.

Le programme principal est très simple. Il se contente d'activer séquentiellement les trois procédures principales du programme :

```
begin
  Presentation; { Présentation du jeu           }
  Jeu;          { Déroulement d'une partie    }
  Fin_Jeu;     { Commentaire de fin de partie }
end.
```

