

9/6

Jeux d'aventures

Si vous aimez les jeux d'aventures ou de rôles, ce chapitre vous concerne. En effet, nous allons étudier une série d'outils qui vous permettront de créer vos propres jeux d'aventures. Nous ferons très souvent allusion aux parties 5 et 6 de l'ouvrage. En effet, des programmes qui nous permettront de construire notre générateur de jeux d'aventures y sont étudiés.

Pour mieux cerner la façon dont va être étudié le générateur d'aventures, posons-nous la question suivante : « Qu'est-ce qu'un jeu d'aventures ? ».

Eh bien, un jeu d'aventures découle d'un scénario qui met en scène un ou plusieurs personnages évoluant dans un ou plusieurs décors. Le joueur participe à l'aventure et choisit son évolution.

Généralement, un seul chemin conduit à l'aboutissement de l'aventure, et plusieurs centaines, voire plusieurs milliers existent... d'où un temps plus ou moins long pour résoudre l'aventure.

Nous voyons donc que, pour écrire un jeu d'aventures, il faudra rédiger un scénario qui décrira :

- l'univers de jeu,
- les possibilités d'intervention du joueur et des actions qui en découlent.

Dans ce chapitre, nous allons voir comment :

- créer l'univers de jeu,
- définir les actions découlant des commandes du joueur,
- agrémenter le jeu d'une musique exécutée sous interruptions,
- créer des jeux et les exécuter.

Remarque

Parfois, il existe une confusion entre les appellations « jeux d'aventures » et « jeux de rôles ».

Les jeux de rôles sont, en fait, des jeux d'aventures dans lesquels le joueur se voit attribuer des caractéristiques et un rôle bien particulier (d'où le nom de jeu de rôle).

I. Création de l'univers de jeu

Au chapitre 10.1 partie 5 nous avons étudié un programme de tracé plein écran qui nous permettait de créer des écrans graphiques en MODE 1. Pour nous limiter dans la consommation en mémoire de tels écrans (16 kilo-octets par écran), nous allons construire une restriction de l'écran normal. L'écran pourra occuper 6 ou 10 kilo-octets.

Entrez les couleurs du dessin et sa taille : petit (6 kilo-octets) ou grand (10 kilo-octets). Le clavier est utilisé de la même manière que pour le programme de tracé en plein écran décrit au chapitre 10.1 partie 5 : (pavé numérique pour le déplacement point par point, touches numériques correspondantes en haut du clavier texte pour le déplacement en « AUTO-REPEAT »).

Le programme de dessin est le suivant :

```

1000 REM *****
1010 REM CONSTITUTION D'ECRANS POUR JEUX D'AVENTURES
1020 REM *****
1030 '
1040 MEMORY &4000
1050 'INTERFACE AVEC MBG ET ABG
1060 FOR I=0 TO 28:READ A:POKE &3018+I,A:NEXT I
1070 DATA &1,&64,&0,&11,&C4,&0,&26,&A,&2E,&A,&3E,&0,&CD,&35,&30,&C9
1080 DATA &1,&64,&0,&11,&C4,&0,&21,&0,&60,&CD,&A6,&30,&C9
1090 '-----
1100 'MBG EN 3035H
1110 FOR I=0 TO 112:READ A:POKE &3035+I,A:NEXT I
1120 DATA &18,&B,&73,&20,&58,&20,&61,&75,&20,&64,&65,&80,&61,&ED,&43,&37,&30,&ED
,&53,&39,&30,&32,&3D,&30,&7C,&32,&3B,&30,&7D,&32,&3C
,&30,&CD,&3,&B9,&21,&0,&60,&11,&94,&2,&3A,&3D,&30,&47,&B7,&28,&4,&ED,&5A,&10,&FC
,&3A,&3B,&30,&77,&23,&22,&3E,&30,&ED,&5B
1130 DATA &37,&30,&2A,&39,&30,&CD,&1D,&BC,&22,&40,&30,&ED,&5B,&3E,&30,&3A,&3B,&3
0,&47,&7E,&12,&23,&13,&10,&FA,&2A,&40,&30,&CD,&26,&B
C,&22,&40,&30,&3A,&3C,&30,&3D,&32,&3C,&30,&20,&E4,&3E,&CF,&12,&ED,&53,&F0,&9C,&C
9
1140 '-----
1150 'ABG EN 30A6H
1160 FOR I=0 TO 72:READ A:POKE &30A6+I,A:NEXT

```

```

1170 DATA &1B,&A,&A,&3B,&D,&A,&4F,&52,&47,&20,&39,&43,&ED,&43,&AB,&30,&ED,&53,&A
A,&30,&7E,&32,&AE,&30,&23,&22,&AC,&30,&CD,&3,&B9,&ED
,&5B,&AB,&30,&2A,&AA,&30,&CD,&1D,&BC,&22,&B0,&30,&ED,&5B,&AC,&30
1180 DATA &3A,&AE,&30,&47,&1A,&77,&23,&13,&10,&FA,&2A,&B0,&30,&CD,&26,&BC,&22,&B
O,&30,&1A,&FE,&CF,&20,&EB,&C9
1190 '-----
1200
1210 'Initialisation
1220 '
1230 STYL0=0 'Style leve
1240 BORDER 0:INK 0,0:INK 1,10:X=0:Y=0
1250 MODE 1
1260 INPUT"Affichage monochrome (O/N)";R#:R#=UPPER$(R#)
1270 IF R#<>"O" AND R#<>"N" THEN 1250
1280 IF R#="O" THEN NBCOUL=1 ELSE NBCOUL=3
1290 PRINT
1300 FOR I=0 TO NBCOUL
1310 PRINT"INK";I;": (0 A 26) ";:INPUT A:INK I,A
1320 NEXT I
1330 PRINT:PRINT "Taille ecran (1=Petit, 2=Grand) ?"
1340 A#=INKEY#:IF A#="" THEN 1340
1350 TE=ASC(A#)-48
1360 IF TE<>1 AND TE<>2 THEN SOUND 1,100,20:GOTO 1340
1370 IF TE=2 THEN POKE &3019,50:POKE &301F,57:POKE &3021,148:POKE &3029,50 ELSE
POKE &3019,75:POKE &301F,44:POKE &3021,124:POKE &302
9,75 'Taille de l'ecran a sauvegarder
1380 PRINT :INPUT"Chargement d'un ecran (O/N) ";R#:R#=UPPER$(R#)
1390 IF R#<>"O" AND R#<>"N" THEN 1380
1400 IF R#="O" THEN PRINT:INPUT"Nom de l'ecran ";ECR#:LOAD ECR#,&6000:CLS:CALL &
3028:GOTO 1420
1410 CLS
1420 IF TE=1 THEN X=152:Y=152:X1=X:Y1=Y:X2=488:Y2=388:PLOT 150,150,1:DRAWR 340,0
:DRAWR 0,240:DRAWR -340,0:DRAWR 0,-240
1430 IF TE=2 THEN X=102:Y=102:X1=X:Y1=Y:X2=538:Y2=388:PLOT 100,100,1:DRAWR 440,0
:DRAWR 0,290:DRAWR -440,0:DRAWR 0,-290

```

```
1440 PLOT X,Y,2
1450 '
1460 'Boucle principale
1470 '
1480 A$=INKEY$:IF A$="" THEN 1480 'Attente action
1490 A=ASC(A$)
1500 IF A<58 AND A>48 THEN ON A-48 GOTO 1510,1520,1530,1540,1440,1550,1560,1570,
1580 ELSE 1590
1510 IF X>X1 AND Y>Y1 THEN PLOT X,Y,STYLO:Y=Y-2:X=X-2:GOTO 1440 ELSE 1620 'En ba
s a gauche
1520 IF Y>Y1 THEN PLOT X,Y,STYLO:Y=Y-2:GOTO 1440 ELSE 1620 'Vers le as
1530 IF X<X2 AND Y>Y1 THEN PLOT X,Y,STYLO:Y=Y-2:X=X+2:GOTO 1440 ELSE 1620 'En ba
s a droite
1540 IF X>X1 THEN PLOT X,Y,STYLO:X=X-2:GOTO 1440 ELSE 1620 'Vers la gauche
1550 IF X<X2 THEN PLOT X,Y,STYLO:X=X+2:GOTO 1440 ELSE 1620 'Vers la droite
1560 IF X>X1 AND Y<Y2 THEN PLOT X,Y,STYLO:Y=Y+2:X=X-2:GOTO 1440 ELSE 1620 'En ha
ut a gauche
1570 IF Y<Y2 THEN PLOT X,Y,STYLO:Y=Y+2:GOTO 1440 ELSE 1620 'Vers le haut
1580 IF X<X2 AND Y<Y2 THEN PLOT X,Y,STYLO:Y=Y+2:X=X+2:GOTO 1440 ELSE 1620 'En ha
ut a droite
1590 IF A=95 OR (A>32 AND A<33+NBCOUL) THEN 1640 'Changement de stylo
1600 IF A=13 THEN 1690 'Fin de trace
1610 GOTO 1480 'Boucle de trace
1620 SOUND 1,100,20:GOTO 1440
1630 '-----
1640 REM Changement de couleur stylo
1650 '
1660 IF A=95 THEN STYLO=0 ELSE STYLO=A-32
1670 GOTO 1440
1680 '-----
1690 REM Fin de trace
1700 '
1710 PLOT X,Y,0:CALL &3018 'Sauvegarde ecran
1720 CLS
```

```
1730 LOCATE 1,10:INPUT"Sauvegarde magnetique (O/N) ";R#:R#=UPPER$(R#)
1740 IF R#<>"O" AND R#<>"N" THEN 1730
1750 IF R#="N" THEN CLS:CALL &3028:GOTO 1440 'Restitution ecran
1760 LOCATE 1,12:INPUT"Nom de l'ecran ";N#
1770 LL=PEEK(&9CF0)+PEEK(&9CF1)*256-&6000+1
1780 SAVE N#,B,&6000,LL 'Sauvegarde
1790 LOCATE 1,14:INPUT"Poursuite (O/N) ";R#:R#=UPPER$(R#)
1800 IF R#="O" THEN CLS:CALL &3028:GOTO 1440 'Restitution ecran
1810 END
```

Lignes 1050 à 1180 : Chargement des sous-programmes ASSEMBLEUR

Lignes 1200 à 1400 : Initialisation du programme

Lignes 1420 à 1430 : Tracé des limites de l'écran

Lignes 1450 à 1610 : Gestion du curseur

Lignes 1660 à 1670 : Changement de la couleur du tracé

Lignes 1710 à 1810 : Fin du tracé avec
sauvegarde d'écran (Ligne 1780)
et/ou Retour au tracé (Ligne 1800).

Remarques :

Les fichiers écran générés par ce programme sont affichables grâce au sous-programme « ABG », contrairement aux fichiers écran générés par le programme de dessin en plein écran étudié au chapitre 10.1 partie 5 qui étaient affichables par la commande LOAD "ECRAN",&C000.

Les sous-programmes Assembleur utilisés ont été décrits au chapitre 10.2 partie 5.

Il s'agit de :

- MBG (Mémorisation de blocs graphiques)
- ABG (Affichage de blocs graphiques)
- Interface BASIC/MBG-ABG

