

## 9/6.5

# Exécution de jeux d'aventures

---

Ce programme permet d'exécuter les jeux d'aventures créés par le précédent.

Une grande partie du programme fait appel aux sous-programmes Assembleur définis dans tout l'ouvrage.

### Utilisation du programme

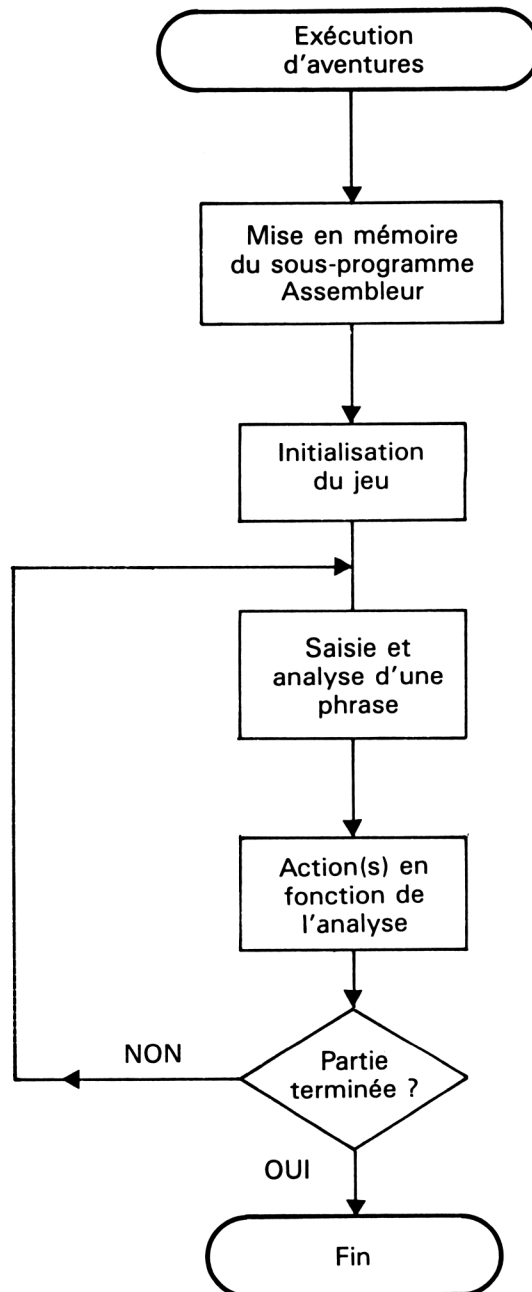
Après avoir poké les données correspondant aux sous-programmes Assembleur (analyse syntaxique d'une phrase, recherche de signification dans l'association de mots clés, chargement d'un écran (ABG), HELP, utilisation de musique sous interruptions), l'ordinateur demande la taille des écrans de jeu (petit ou grand), et le nom de l'aventure à exécuter. Pour la bonne suite des opérations, positionnez vous en majuscules (CAPS LOCK) si nécessaire (même mode que pour la définition des mots clés dans le programme de création de jeux d'aventures décrit précédemment).

L'aventure est alors chargée, et le premier écran apparaît avec le texte de présentation dans une fenêtre de texte de trois lignes situées en bas de l'écran.

Si l'option **Musique** a été choisie dans le générateur d'aventures, vous pouvez l'entendre dès le début de la partie.

Si l'option **HELP** a été choisie dans le générateur d'aventures, l'appui simultané sur les touches **Ctrl** et **H** vous permet d'avoir la liste des mots clés. Les touches-flèches vers le haut et vers le bas font défiler les mots clés dans la fenêtre de texte. L'appui sur **Ctrl H** vous fait revenir au mode **Entrée**.

La logique du programme est la suivante :



Le listing du programme est le suivant :

```

1000 REM EXECUTION D'AVENTURES
1010 MEMORY &5000
1020 GOSUB 1040 'Initialisation et chargement du jeu
1030 GOSUB 2580 'Jeu
1040 '=====
=====
1050 'S/P en ASSEMBLEUR
1060 '=====
=====
1070 'Analyse d'une phrase
1080 '=====
=====
1090 FOR i=&9008 TO &907E
1091   READ a$
1092   a=VAL("&" + a$)
1093   POKE i, a
1094 NEXT i
1101 DATA 2A,4,90,22,6,90,3E,1,32,0,90,2A,2,90,7E,32
1102 DATA 1,90,23,22,2,90,CD,65,90,E,0,C5,2A,2,90,ED
1103 DATA 5B,4,90,CD,70,90,B7,C1,C8,C,79,FE,20,2B,9,2A
1104 DATA 4,90,23,22,4,90,1B,F3,3A,1,90,3D,32,1,90,2B
1105 DATA 16,2a,6,90,22,4,90,2a,2,90,cd,65,90,23,3a,0
1106 DATA 90,3c,32,0,90,1B,bc,3e,ff,32,0,90,c9,6,0,5
1107 DATA 4,7e,fe,ff,C8,23,1B,FB,1A,BE,20,6,23,13,10,FB
1108 DATA 1B,3,3E,1,C9,AF,C9
1120 '=====
=====
1130 'Recherche de verbe, sujet et relation entre eux
1140 '=====
=====
1150 FOR i=&90BD TO &9102
1151   READ a$
1152   a=VAL("&" + a$)
1153   POKE i, a
1154 NEXT i
1161 DATA 21,89,90,AF,6,4,77,23,10,FC,2A,83,90,22,2,90
1162 DATA 21,0,43,22,4,90,CD,8,90,3A,0,90,32,8B,90,2A
1163 DATA 85,90,22,2,90,21,0,43,22,4,90,CD,8,90,3A,0
1164 DATA 90,32,87,90,2A,81,90,7E,32,80,90,3A,7F,90,23,BE
1171 DATA 20,24,23,3A,8B,90,BE,20,1E,23,3A,87,90,BE,20,1B
1172 DATA 23,7E,32,89,90,23,7E,32,8A,90,23,7E,32,8B,90,23,7E
1173 DATA 32,8C,90,1B,F,23,23,23,23,23,23,3A,80,90,3D,32
1174 DATA 80,90,20,C6,C9
1180 '=====
=====
1190 'Activation de INPUT et analyse de la reponse du joueur
1200 '=====
=====

```

```

1210 FOR i=&91A1 TO &91B5
1211   READ a$
1212   a=VAL("&" + a$)
1213   POKE i, a
1214 NEXT i
1221 DATA 21,17,1,3E,26,11,0,43,CD,45,93,3A,A0,91,32,7F
1222 DATA 90,CD,8D,90,C9
1230 '=====
=====
1240 'Chargement d'un ecran
1250 '=====
=====
1260 FOR i=&91B6 TO &91C2
1261   READ a$
1262   a=VAL("&" + a$)
1263   POKE i, a
1264 NEXT i
1270 FOR i=&91C3 TO &920B
1271   READ a$
1272   a=VAL("&" + a$)
1273   POKE i, a
1274 NEXT i
1281 DATA 1,64,0,11,CO,0,21,0,60,CD,C3,91,C9,18,A,A
1291 DATA 3B,D,A,4F,52,47,20,39,43,ED,43,AB,30,ED,53,AA
1292 DATA 30,7E,32,AE,30,23,22,AC,30,CD,3,B9,ED,5B,AB,30
1293 DATA 2A,AA,30,CD,1D,BC,22,BO,30,ED,5B,AC,30,3A,AE,30
1294 DATA 47,1A,77,23,13,10,FA,2A,BO,30,CD,26,BC,22,BO,30
1295 DATA 1A,FE,CF,20,EB,C9
1310 '=====
=====
1320 'Fonction HELP
1330 '=====
=====
1340 FOR i=&9211 TO &932B
1341   READ a$
1342   a=VAL("&" + a$)
1343   POKE i, a
1344 NEXT i
1351 DATA 21,16,0,11,18,27,CD,66,BB,CD,6C,BB,CD,84,BB,CD
1352 DATA 6C,BB,21,0,80,22,C,92,21,17,1,CD,75,BB,2A,C
1353 DATA 92,CD,FE,92,2A,C,92,23,22,C,92,22,F,92,21,18
1354 DATA 1,CD,75,BB,2A,C,92,CD,FE,92,2A,C,92,23,22,C
1355 DATA 92,21,19,1,CD,75,BB,2A,C,92,CD,FE,92,3E,0,32
1356 DATA E,92,26,0,16,22,2E,16,1E,19,D5,E5,2A,F,92,22
1357 DATA C,92,CD,6,BB,FE,8,28,65,FE,F0,28,6,FE,F1,28
1358 DATA 2D,18,EF,3A,E,92,FE,A,28,E8,3C,32,E,92,6,1
1359 DATA E1,D1,3E,0,D5,E5,CD,50,BC,CD,E5,92,CD,E5,92,CD
1361 DATA F5,92,CD,1C,93,CD,1C,93,23,22,C,92,18,C4,3A,E
1362 DATA 92,FE,0,28,BD,3D,32,E,92,6,0,E1,D1,3E,0,D5
1363 DATA E5,CD,50,BC,2A,C,92,2B,22,C,92,CD,1C,93,CD,1C
1364 DATA 93,23,22,C,92,CD,11,93,23,22,C,92,18,94,D1,E1

```

```

1365 DATA CD,6C,BB,C9,2A,C,92,7E,FE,FF,28,3,23,18,F8,23
1366 DATA 22,C,92,C9,21,3,1,CD,75,BB,2A,C,92,7E,FE,FF
1367 DATA 28,A,FE,20,38,3,CD,5A,BB,23,18,F1,22,C,92,C9
1368 DATA 21,1,1,CD,75,BB,2A,C,92,18,E2,2A,C,92,2B,7E,FE
1369 DATA FF,28,3,2B,18,F8,22,C,92,C9
1400 '=====
=====
1410 'Fonctions utilitaires: LOCATE+INPUT
1420 '=====
=====
1430 FOR I=&9345 TO &93CB
1431   READ a#
1432   a=VAL("&" + a#)
1433   POKE i,a
1434 NEXT i
1440 DATA 32,44,93,22,3F,93,CD,7B,BB,CD,81,BB,CD,75,BB,6
1441 DATA 0,2A,3F,93,CD,18,BB,FE,7F,28,26,FE,D,28,4F,FE
1442 DATA 8,28,55,32,43,93,3A,44,93,B8,38,E8,C5,D5,E5,3A
1443 DATA 43,93,CD,5D,BB,E1,D1,C1,4,24,3A,43,93,12,13,18
1444 DATA D3,22,41,93,D5,EB,2A,3F,93,7A,BC,28,28,D1,5,2A
1445 DATA 41,93,25,22,41,93,CD,75,BB,C5,D5,3E,20,CD,5D,BB
1446 DATA D1,C1,2A,41,93,CD,75,BB,2A,41,93,1B,18,A6,3E,FF
1447 DATA 12,CD,84,BB,C9,D1,1B,9C,32,43,93,3A,CC,93,B7,28
1448 DATA A5,3E,1,32,CD,93,C9
1470 '=====
=====
1480 'Execution de fichiers musicaux sous IT
1490 '=====
=====
1500 FOR i=&A000 TO &A0D5
1501   READ a#
1502   a=VAL("&" + a#)
1503   POKE i,a
1504 NEXT i
1521 DATA C3,2F,A0,1,0,0,0,0,0,0,16,0,1,1,0,0
1522 DATA 0,0,0,16,0,3,2,7,1,1,0,2,5,FF,4,0
1523 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,21
1524 DATA 25,A0,6,B1,E,0,11,49,A0,CD,EF,BC,21,1F,A0,11
1525 DATA 1,0,1,B,0,CD,E9,BC,C9,F3,F5,C5,D5,E5,DD,E5
1526 DATA FD,E5,3A,2E,A0,B7,20,15,3E,1,32,2E,A0,3E,1,21
1527 DATA 15,A0,CD,BC,BC,FD,21,D3,A0,FD,22,2C,A0,FD,2A,2C
1528 DATA A0,E,1,CD,A0,A0,E,4,CD,A0,A0,E,2,CD,A0,A0
1529 DATA FD,7E,0,FE,FF,20,B,FD,7E,1,FE,FF,20,4,FD,21
1531 DATA D3,A0,FD,22,2C,A0,FD,E1,DD,E1,E1,D1,C1,F1,FB,C9
1532 DATA FD,7E,0,B7,20,F,FD,7E,1,B7,20,9,DD,21,3,A0
1533 DATA DD,71,0,18,13,DD,21,C,A0,DD,71,0,FD,7E,0,DD
1534 DATA 77,3,fd,7E,1,DD,77,4,DD,E5,E1,CD,AA,BC,FD,23
1535 DATA FD,23,C9,0,0,0

```

```

1550 '=====
=====
1560 'Presentation
1570 '=====
=====
1580 INK 0,0:BORDER 0:INK 1,10:INK 2,1,3:MODE 1:PEN 2
1590 DIM T1(100):BIS=1
1600 EF$="" " 'Effacement de ligne
1610 PRINT" EXECUTION D'AVENTURE":PEN 1
1620 LOCATE 4,10:PRINT"Ecran: P)etit ou G)rand : ";
1630 A$=INKEY$:IF A$="" THEN 1630
1640 A$=UPPER$(A$)
1650 IF A$<>"P" AND A$<>"G" THEN 1630
1660 PRINT A$:IF A$="P" THEN POKE &91B7,75 ELSE POKE &91B7,50
1670 LOCATE 4,18:INPUT"Nom de l'aventure ";A$
1680 LOAD A$,&8000 'Chargement de l'aventure
1690 '=====
=====
1700 'Memorisation de l'aventure et
1710 'Interfacege avec l'assembleur
1720 '=====
=====
1730 AD=&8000 'Adresse de debut d'aventure
1740 SD=PEEK(AD) 'Salle de depart
1750 AD=AD+2:TD$=""
1760 IF PEEK(AD)<>&FF THEN TD$=TD$+CHR$(PEEK(AD)):AD=AD+1:GOTO 1760 'Texte de depart
1770 AD=AD+1:MUS$="":IF PEEK(AD)=&FF THEN 2310 'Pas de musique
1780 IF PEEK(AD)<>&FF THEN MUS$=MUS$+CHR$(PEEK(AD)):AD=AD+1:GOTO 1780 'Nom du fichier musical
1790 IF MUS$="" OR BIS=0 THEN 2310 'Pas de musique pendant le jeu
1800 'Chargement d'un morceau
1810 '
1820 OPENIN MUS$:BIS=0
1821 DIM NO$(30,3)
1830 INPUT #9,NB 'Nombre de notes
1840 FOR I=0 TO NB-1
1850   FOR J=1 TO 3
1860     INPUT #9,NO$(I,J)
1870   NEXT J
1880 NEXT I
1890 CLOSEIN
1900 FOR I=1 TO 3 'Recherche du nombre de notes sur chaque voie
1910   X=NB
1920   WHILE NO$(X,I)=""
1930     X=X-1

```

```

1940 WEND
1950 N(I)=X+1
1960 NEXT I
1970 FOR I=1 TO 96 'Chargement des notes pour conversion
1980 READ T1(I)
1990 NEXT I
2000 DATA 3822,3608,3405,3214,3034,2863,2703,2551,2408,2273,
2145,2025
2010 DATA 1911,1804,1703,1607,1517,1432,1351,1276,1204,1136,
1073,1012
2020 DATA 956,902,851,804,758,716,676,638,602,568,536,506
2030 DATA 478,451,426,402,379,358,338,319,301,284,268,253
2040 DATA 239,225,213,201,190,179,169,159,150,142,134,127
2050 DATA 119,113,106,100,95,89,84,80,75,71,67,63
2060 DATA 60,56,53,50,47,45,42,40,38,36,34,32
2070 DATA 30,28,27,25,24,22,21,20,19,18,17,16
2080 '
2090 AN=&A0D3 'Adresse 1ere note a jouer
2100 A=0
2110 FOR I=1 TO 3
2120 IF N(I)>A THEN A=N(I)
2130 NEXT I
2140 FOR I=0 TO A-1
2150 FOR J=1 TO 3
2160 IF NO$(I,J)="" THEN MSB=0:LSB=0:GOTO 2250 'Rien a j
ouer
2170 L$=LEFT$(NO$(I,J),2)
2180 L=- (L$="DO") - (L$="RE") *3 - (L$="MI") *5 - (L$="FA") *6 - (L
$="SO") *8 - (L$="LA") *10 - (L$="SI") *12
2190 IF MID$(L$,3,1)="#" THEN L=L+1
2200 IF MID$(L$,3,1)="B" THEN L=L-1
2210 R$=RIGHT$(NO$(I,J),1):R=ASC(R$)-48
2220 L=(R-1)*12+L
2230 NT=T1(L) 'Note a jouer
2240 MSB=INT(NT/256):LSB=NT-MSB*256
2250 POKE AN,LSB:AN=AN+1:POKE AN,MSB:AN=AN+1
2260 NEXT J
2270 NEXT I
2280 POKE AN,&FF:AN=AN+1:POKE AN,&FF
2290 CALL &A02F 'Activation musique
2300 '
2310 AD=AD+1:HELP=PEEK(AD):POKE &93CC,HELP 'Validation touch
e HELP
2320 AD=AD+2:EC=AD 'Pointeur sur debut ecrans
2330 GOSUB 2390:VE=AD+2^16:V1=VE-INT(VE/256)*256:v2=INT(ve/2
56):POKE &9083,v1:POKE &9084,v2:POKE &9224,v1:POKE &9225,v2
'Pointeur sur debut verbes et interface avec help
2340 GOSUB 2390:SU=AD+2^16:POKE &9085,SU-INT(SU/256)*256:POK
E &9086,INT(SU/256) 'Pointeur sur debut sujets
2360 GOSUB 2390:RE=AD 'Pointeur sur debut reponses

```

```

2370 GOSUB 2390:CO=AD+2^16:POKE &9081,CO-INT(CO/256)*256:POKE
&9082,INT(CO/256) 'Pointeur sur debut conditions
2371 L1=PEEK(VE):L2=PEEK(SU):L3=L1+L2-3:IF L3>=0 THEN POKE &
9288,L3 ELSE POKE &9288,0 'Nombre max de mots a afficher -3
2380 RETURN
2390 A=PEEK(AD)+1
2400 A=A-1:AD=AD+1:IF A=0 THEN 2430
2410 AD=AD+1:IF PEEK(AD)<>&FF THEN 2410
2420 GOTO 2400
2430 RETURN
2440 '*****
*****
2450 'Lecture d'une variable en memoire
2460 A$=""
2470 IF PEEK(AD)<>&FF THEN A$=A$+CHR$(PEEK(AD)):AD=AD+1:GOTO
2470
2480 RETURN
2490 '*****
*****
2500 'Chargement d'ecran
2510 '
2520 GOSUB 2400 'Recherche du debut du nom de l'ecran
2530 A$="" 'Nom de l'ecran
2540 IF PEEK(AD)<>&FF THEN A$=A$+CHR$(PEEK(AD)):AD=AD+1:GOTO
2540
2550 POKE &A00D,0:LOAD A$:POKE &A00D,1:CALL &91B6 'Chargemen
t avec devalidation puis revalidation de la musique
2560 RETURN
2570 '-----
-----

2580 REM Initialisation du jeu
2590 PTS=200 'Points en debut de partie
2600 CLS:A=$D:POKE &91A0,A:AD=EC:GOSUB 2500 'Recherche du no
m de l'ecran et chargement
2610 LOCATE 1,22:PRINT TD$ 'Texte de depart
2630 '-----
-----

2640 REM Boucle principale de jeu
2650 POKE &93CD,0 'CTRL+H Non active
2660 CALL &91A1 'INPUT+Analyse reponses
2670 IF PEEK(&93CD)=1 THEN CALL &9211:WINDOW 1,40,1,25:GOTO
2640 'Activation du HELP
2680 POKE &93CD,0 'CTRL+H non active
2690 P1=PEEK(&9089):P2=PEEK(&908A):P3=PEEK(&908B):P4=PEEK(&9
08C):IF P1+P2+P3+P4=0 THEN LOCATE 1,22:PRINT EF$:LOCATE 1,22
:PRINT"Pardon ?":GOTO 2640
2700 IF P1<>0 THEN A=P1:POKE &91A0,A:AD=EC:GOSUB 2500 'Chang
ement de salle
2710 LOCATE 1,22:PRINT EF$ 'Effacement texte de presentation
2720 LOCATE 1,23:PRINT EF$ 'Effacement ligne texte

```



```
2730 IF P2<>0 THEN A=P2:AD=RE:GOSUB 2400:GOSUB 2460:LOCATE 1
,22:PRINT A$ 'Reponse
2740 IF P3<>0 THEN PTS=PTS+P3-128:LOCATE 35,1:PRINT PTS 'Nom
bre de points
2750 IF PTS<=0 THEN 2770 'Fin de partie
2760 IF P4=0 THEN 2800
2770 LOCATE 1,22:PEN 2:PRINT"Fin de partie":PEN 1:INPUT"On r
ecommence (O/N)";R$
2780 R$=UPPER$(R$):IF R$<>"O" AND R$<>"N" THEN 2770
2790 IF R$="O" THEN GOSUB 1730:GOSUB 2580 ELSE END 'Nouvelle
aventure ou fin
2800 GOTO 2640
```

- Lignes 1090 à 1540 : Initialisation.
- Lignes 1580 à 1680 : Chargement des sous-programmes Assembleur.  
Choix de l'aventure à exécuter.
- Ligne 1680 : Chargement de l'aventure.
- Lignes 1730 à 2430 : Mémorisation de l'aventure.  
(Transfert du fichier « aventure » dans des tableaux utilisés en Basic.)
- Lignes 2500 à 2560 : Chargement d'un écran et affichage.
- Lignes 2580 à 2610 : Initialisation d'une partie
- Lignes 2640 à 2800 : Boucle de jeu.

